Application Note AN-008

BusTools/1553 Error Injection and Detection

abaco
SYSTEMS

Copyrights

Abaco Systems, Inc.
26 Castilian Drive, Suite B
Goleta, CA 93117
Main   +1 805-965-8000 or +1 805-883-6101
Support +1 805-965-8000 or +1 805-883-6097

support@abaco.com (email)
https://www.abaco.com/products/avionics

Additional Resources

For more information, please visit the Abaco Systems website at:

www.abaco.com
support@condoreng.com (e-mail)
http://www.condoreng.com (Internet)

Contents

Introduction

Error injection and the reporting of errors on the bus is one of the more confusing (and often misunderstood) aspects of the 1553 bus. The purpose of this document is to explain how we do error injection with BusTools/1553 and to explain why the bus monitor reports what it does for various error conditions.

The examples shown in this document were done with BusTools/1553 version 6.40 on a PCCARD-D1553 board. A Tektronix TDS1012 oscilloscope was used to view the messages and words on the bus.

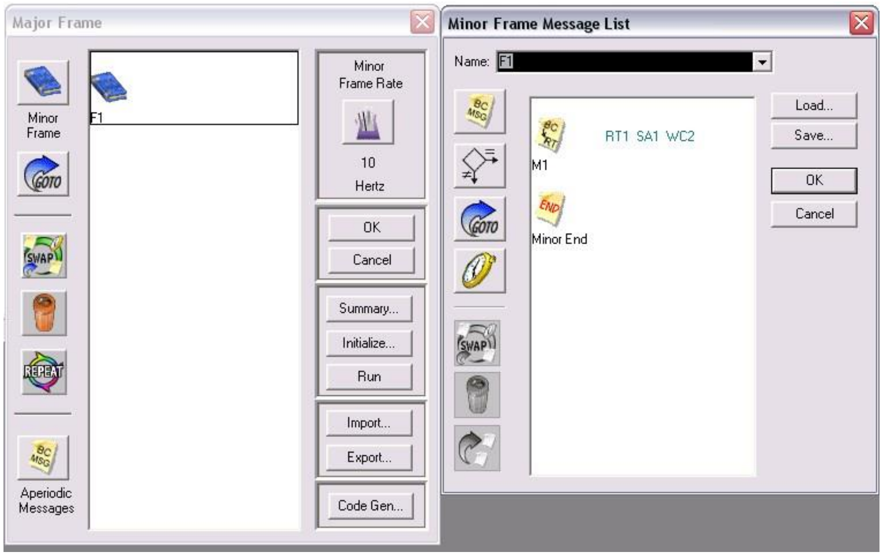
Application Note 003 explains how to use an oscilloscope with the 1553 bus.

This Application Note assumes a basic knowledge of 1553. For information on the MIL-STD-1553 protocol refer to the “MIL-STD-1553 Tutorial” document available from Condor Engineering Inc. Additional information can also be found in MIL-HDBK-1553A.

Error Injection from the Bus Controller

The Bus Controller can inject errors on COMMAND words and (for receive messages) on DATA words.

We will setup a very simple configuration to use for our examples. Here we setup the BC to send a single message (RT1 RECEIVE SA1 WC2). We set the data words to random values.



Name: F1

Minor Frame Rate 10 Hertz

M1 RT1 SA1 WC2

Errors on COMMAND Words

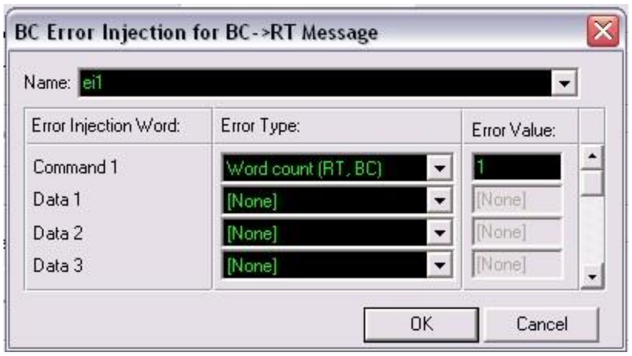
The Bus Controller can inject errors on COMMAND words. **However, when there is an error on the command word the Bus Monitor will not see the message at all.** This is because without a valid command word there is no message – the bad command word and any data words are ignored as noise or other spurious data on the bus. In cases like this you will need to use an oscilloscope to see the signal on the bus. Even though the Bus Monitor reports nothing, the error injection does work and the bad command does go out on the bus (but the decoder ignores it because there is no valid command to start message processing).

Word Count Errors

Word count errors cause the BC to send a number of words different from the word count specified in the command word.

Too Few Words

Here we setup an error where we send only one word rather than the expected two words.

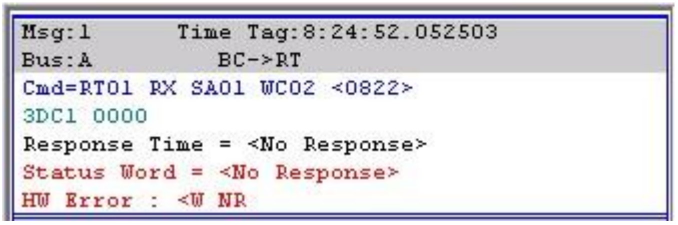


BC Error Injection for BC->RT Message

Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	Word count (RT, BC)	1
Data 1	[None]	[None]
Data 2	[None]	[None]
Data 3	[None]	[None]

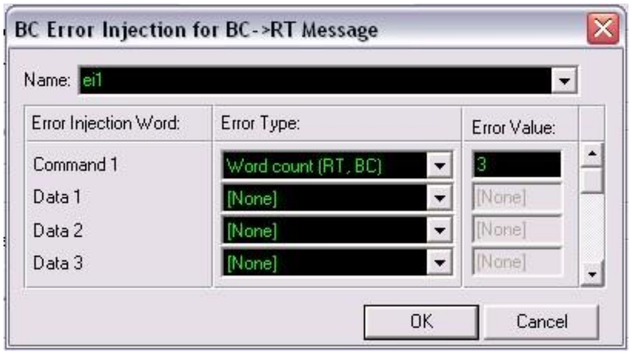
The BM reports the following:



```
Msg:1       Time Tag:8:24:52.052503
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 0000
Response Time = <No Response>
Status Word = <No Response>
HW Error : <W NR
```

Too Many Words

Here we setup an error with three words rather than the expected two words.

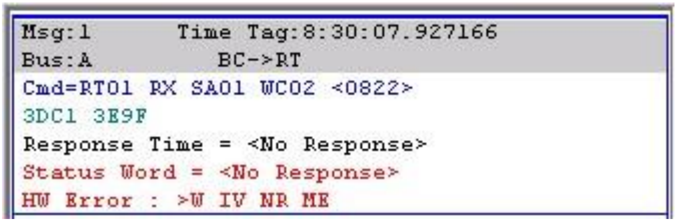


BC Error Injection for BC->RT Message

Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	Word count (RT, BC)	3
Data 1	[None]	[None]
Data 2	[None]	[None]
Data 3	[None]	[None]

The BM reports the following:

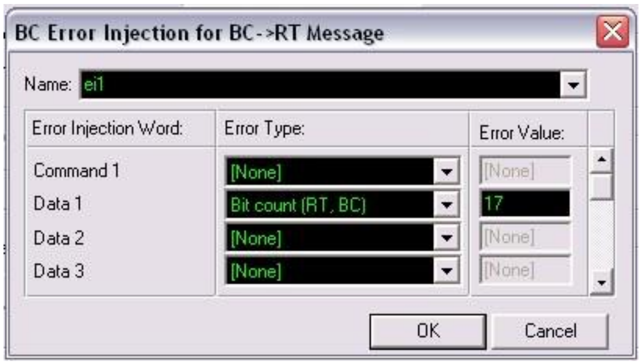


```
Msg:1       Time Tag:8:30:07.927166
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 3E9F
Response Time = <No Response>
Status Word = <No Response>
HW Error : >W IV NR ME
```

Bit Count Errors

A normal word consists of 16 bits. A bit count error causes the board to send a specified word with an incorrect number of bits. Bit count errors can be injected on COMMAND or DATA words.

Here we setup a bit count error on the first data word. In this case we will have 17 bits rather than the expected 16 bits.



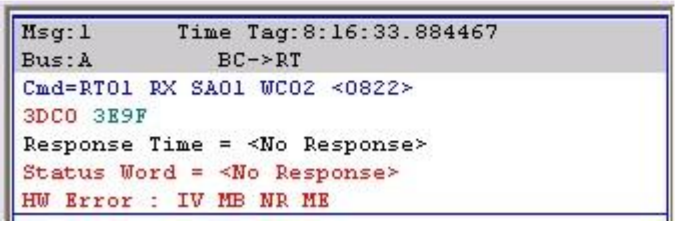
BC Error Injection for BC->RT Message

Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	[None]	[None]
Data 1	Bit count (RT, BC)	17
Data 2	[None]	[None]
Data 3	[None]	[None]

The valid values for the bit count error are 14, 15, 17, 18, or 19 (14-19, not including 16). We will show the results of using bit counts 15, 17, and 19.

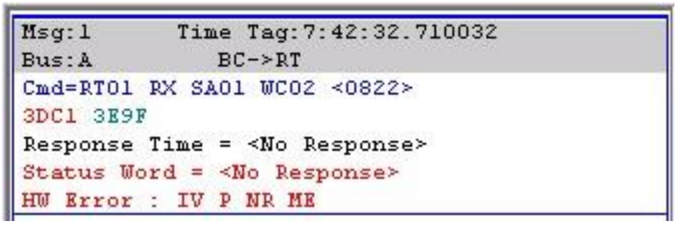
Below is what the BM reported when we had one bit too few (15):



```
Msg:1       Time Tag:8:16:33.884467
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC0 3E9F
Response Time = <No Response>
Status Word = <No Response>
HW Error : IV MB NR ME
```

The BM reports invalid word, mid-bit zero crossing error, no response, and message error. In this case we have one bit too few, so the sync pattern of the following word is seen where the parity bit should be – this means that there is no zero crossing at the mid-bit point of the parity bit, so we get a mid-bit zero crossing error.

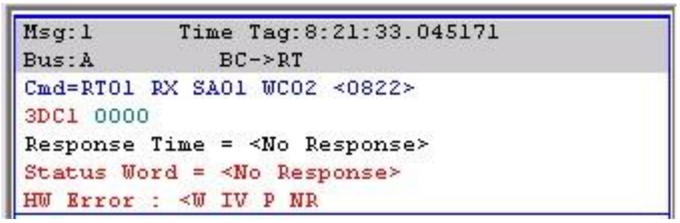
Below is what the BM reported when we had one extra bit (17):



```
Msg:1       Time Tag:7:42:32.710032
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 3E9F
Response Time = <No Response>
Status Word = <No Response>
HW Error : IV P NR ME
```

The BM reports the following errors – invalid word, parity error, no response, and message error. In this case we have injected one extra bit in the word – this pushes out the parity bit and so a parity error is seen.

Below is what the BM reported when we had three extra bits (19):

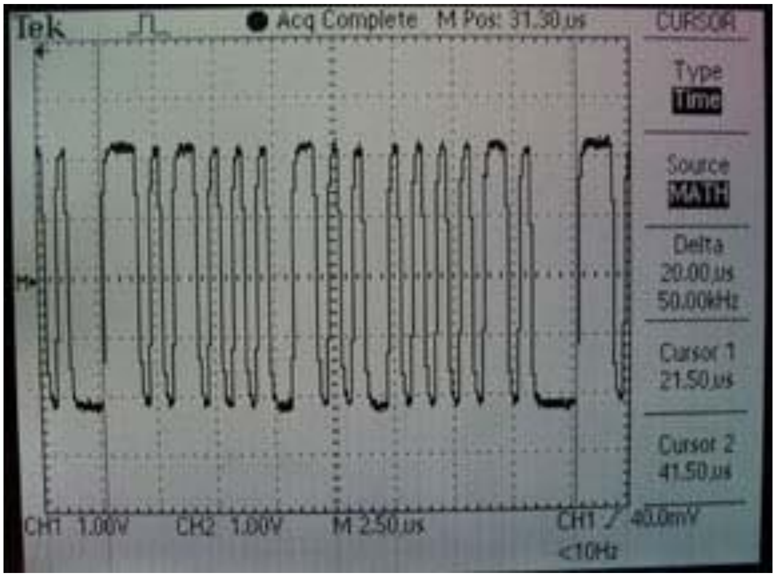


```
Msg:1       Time Tag:8:21:33.045171
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 0000
Response Time = <No Response>
Status Word = <No Response>
HW Error : <W IV P NR
```

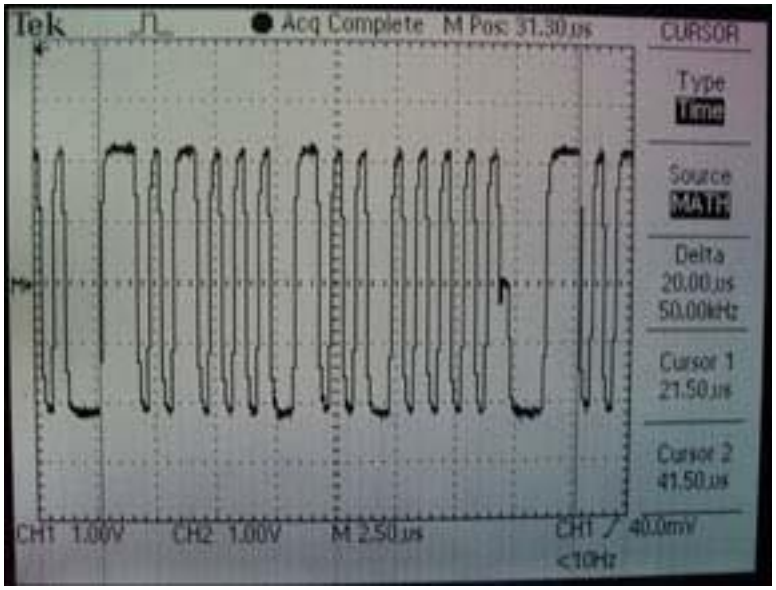
If we inject three extra bits (19), this makes the word long enough that the sync is lost for the next word, so a low word count error is reported (<W).

This demonstrates how one error can be reported in multiple ways (parity error, zero crossing error, word count error, etc.), AND ALL OF THESE ARE CORRECT – the only way to completely understand what happened on the bus is to look at it with an oscilloscope. We show below what was seen on the oscilloscope for each of these cases (looking at the first data word):

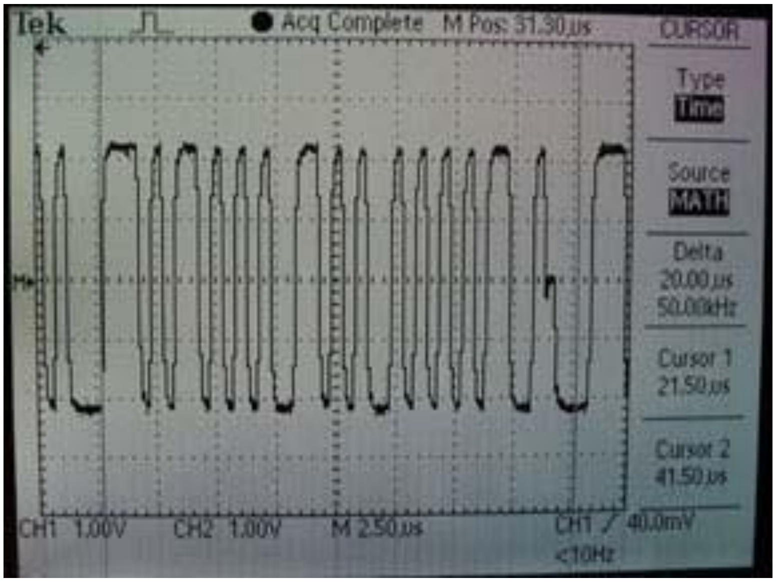
Normal condition – no error:



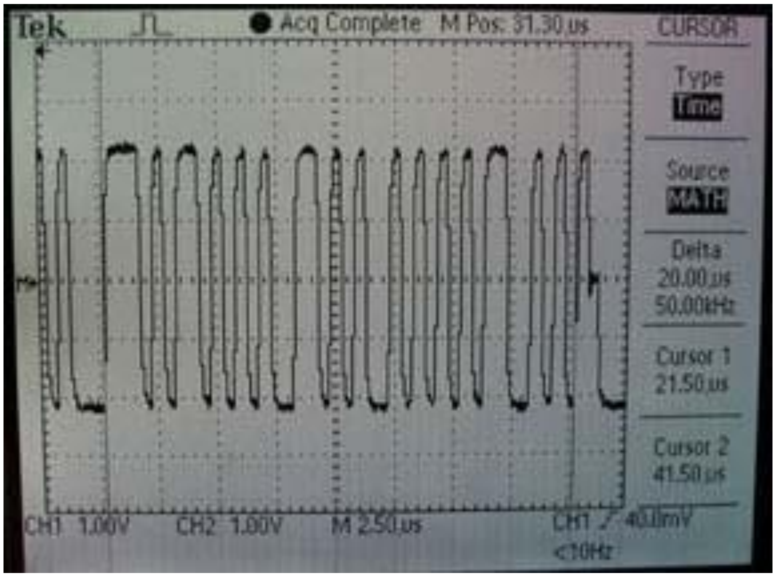
Too few bits (15):



Too many bits (17):

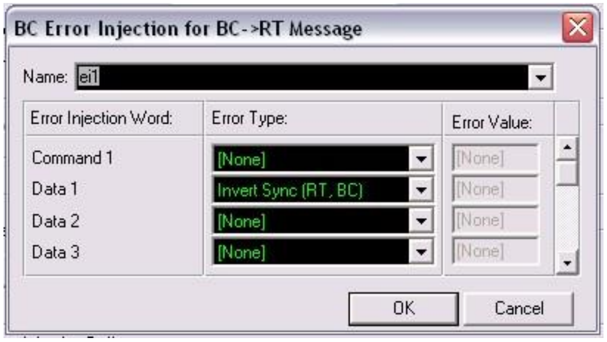


Too many bits (19):



Inverted Sync Errors

Here we setup an inverted sync error on the first data word.

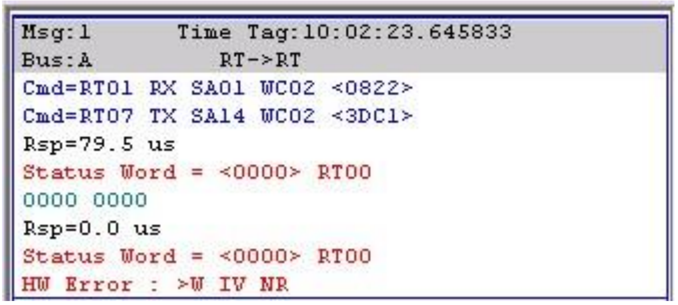


BC Error Injection for BC->RT Message

Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	[None]	[None]
Data 1	Invert Sync (RT, BC)	[None]
Data 2	[None]	[None]
Data 3	[None]	[None]

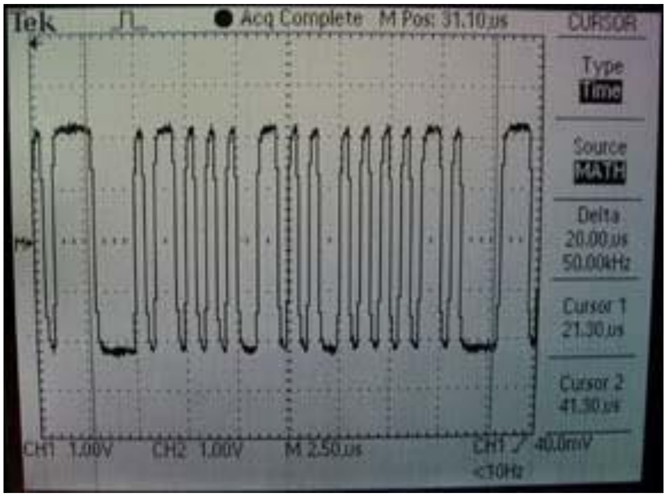
This is what the BM reports:



```
Msg:1       Time Tag:10:02:23.645833
Bus:A          RT->RT
Cmd=RT01 RX SA01 WC02 <0822>
Cmd=RT07 TX SA14 WC02 <3DC1>
Rsp=79.5 us
Status Word = <0000> RT00
0000 0000
Rsp=0.0 us
Status Word = <0000> RT00
HW Error : >W IV NR
```

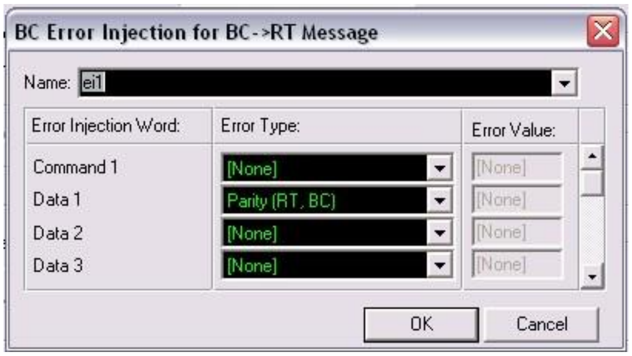
At first this may seem like nonsense. However, in this case the first data word has an inverted sync, which means it now has a COMMAND sync. Therefore the first data word looks like a command word contiguous with the original command word, which looks just like an RT-RT message! Therefore this is correct.

Here is what we see on the oscilloscope for the first data word – note the COMMAND sync pattern:



Parity Errors

When we inject a PARITY ERROR, this just inverts the parity bit on the selected word. Here we inject a parity error on the first data word.

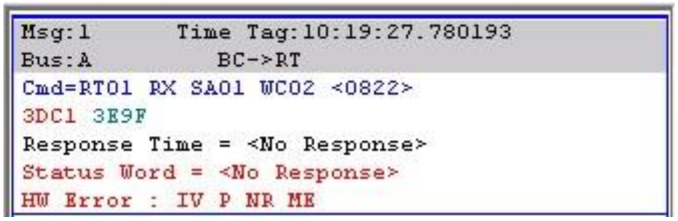


BC Error Injection for BC->RT Message

Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	[None]	[None]
Data 1	Parity (RT, BC)	[None]
Data 2	[None]	[None]
Data 3	[None]	[None]

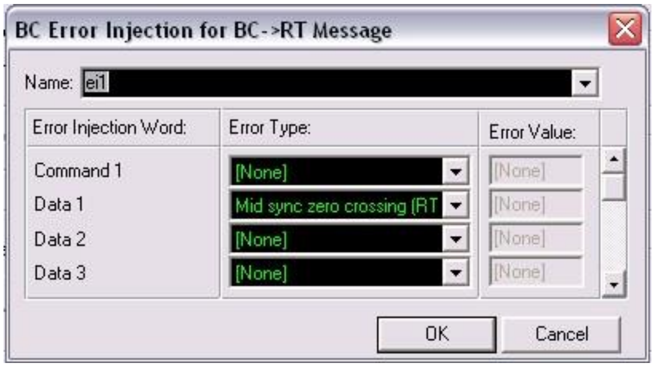
This is what the BM reports:



```
Msg:1       Time Tag:10:19:27.780193
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 3E9F
Response Time = <No Response>
Status Word = <No Response>
HW Error : IV P NR ME
```

Mid-Sync Zero Crossing Errors

In this case we cause the zero crossing in the middle of the sync pattern to be late on data word 1.

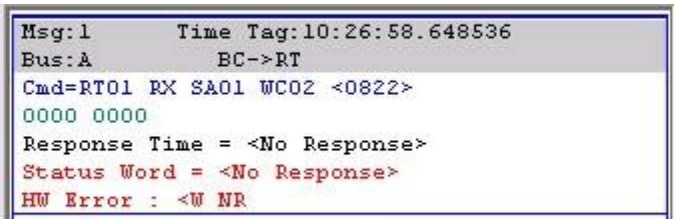


BC Error Injection for BC->RT Message

Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	[None]	[None]
Data 1	Mid sync zero crossing (RT	[None]
Data 2	[None]	[None]
Data 3	[None]	[None]

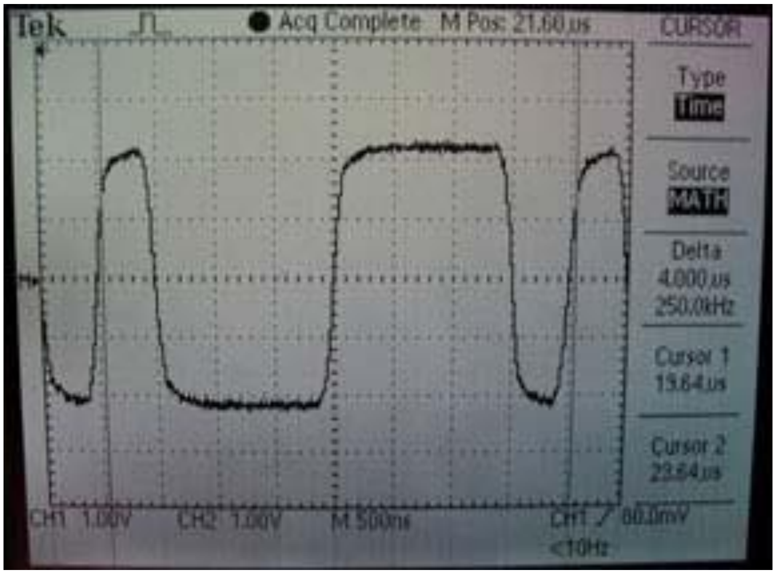
This is what the BM reports:



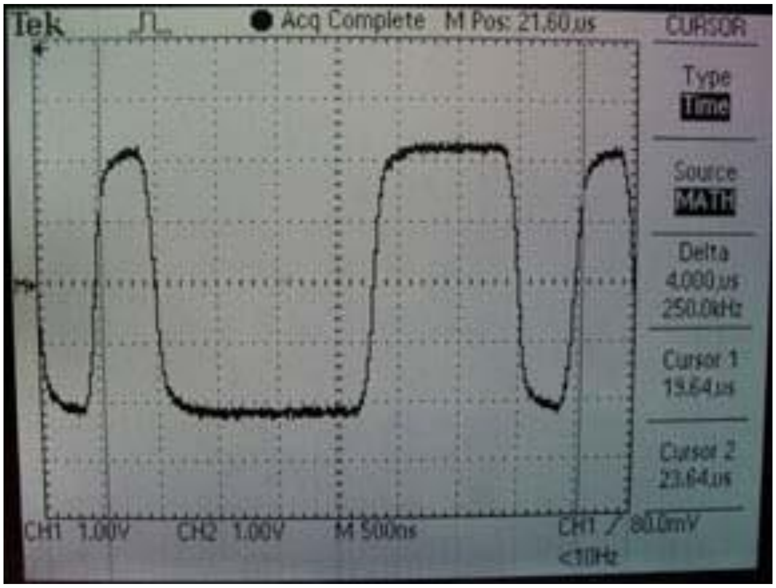
```
Msg:1       Time Tag:10:26:58.648536
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
0000 0000
Response Time = <No Response>
Status Word = <No Response>
HW Error : <W NR
```

Because the first data word has a bad sync pattern, it is not recognized as a valid data word, so the bus monitor reports “<W” indicating too few words in the message.

Let’s look at it with the oscilloscope. First, here is the sync pattern of the first data word with no error:

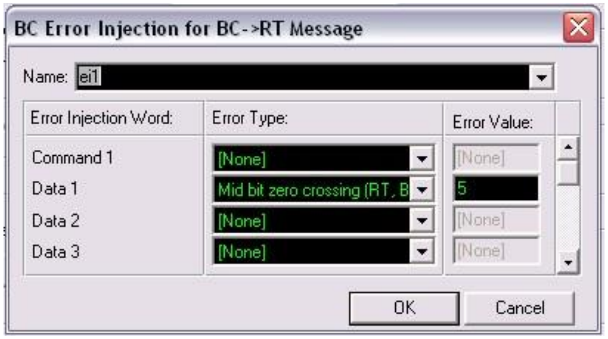


Here it is with the mid-sync zero crossing error:



Mid-Bit Zero Crossing Errors

In this case, we create a zero crossing error on a bit in the word (rather than on the sync pattern). Here we inject a mid-bit zero crossing error on bit 5 of data word 1.

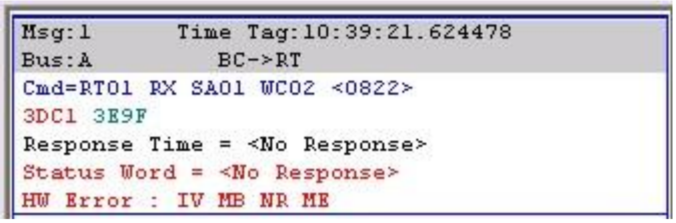


BC Error Injection for BC->RT Message

Name: ei1

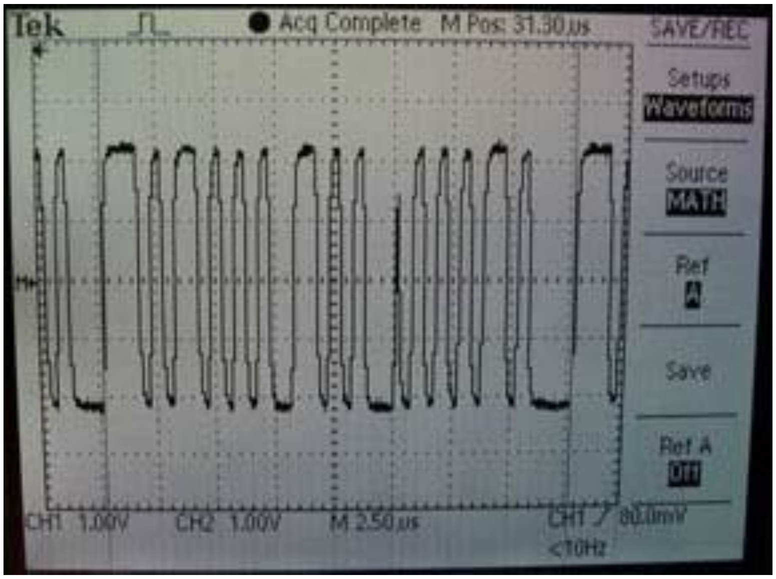
Error Injection Word:	Error Type:	Error Value:
Command 1	[None]	[None]
Data 1	Mid bit zero crossing (RT, B	5
Data 2	[None]	[None]
Data 3	[None]	[None]

This is what the BM reports:



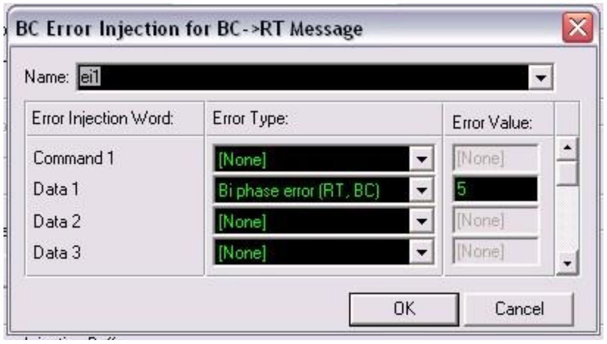
```
Msg:1       Time Tag:10:39:21.624478
Bus:A          BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 3E9F
Response Time = <No Response>
Status Word = <No Response>
HW Error : IV MB NR ME
```

Here is what we see on the oscilloscope:



Bi-Phase Encoding Errors

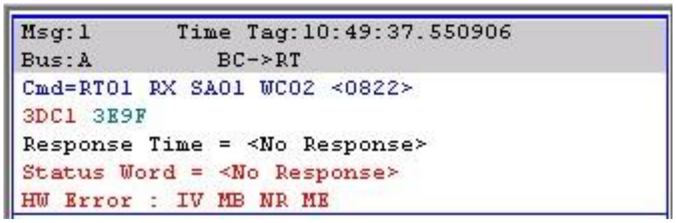
A bi-phase encoding error means that a zero crossing was missed entirely. Here we setup a bi-phase error on bit 5 of data word 1.


BC Error Injection for BC->RT Message

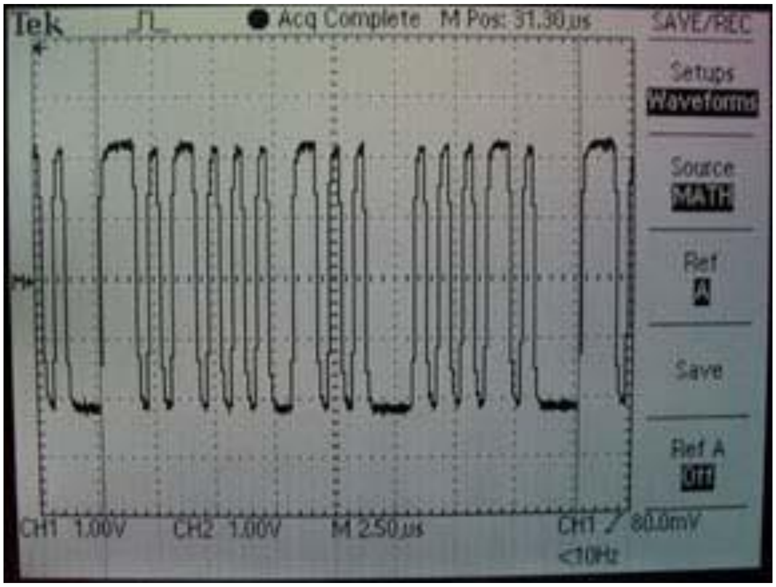
Name: ei1

Error Injection Word:	Error Type:	Error Value:
Command 1	[None]	[None]
Data 1	Bi phase error (RT, BC)	5
Data 2	[None]	[None]
Data 3	[None]	[None]

This is what the BM reports:


```
Msg:1       Time Tag:10:49:37.550906
Bus:A            BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 3E9F
Response Time = <No Response>
Status Word = <No Response>
HW Error : IV MB NR ME
```

Here is what we see on the oscilloscope:

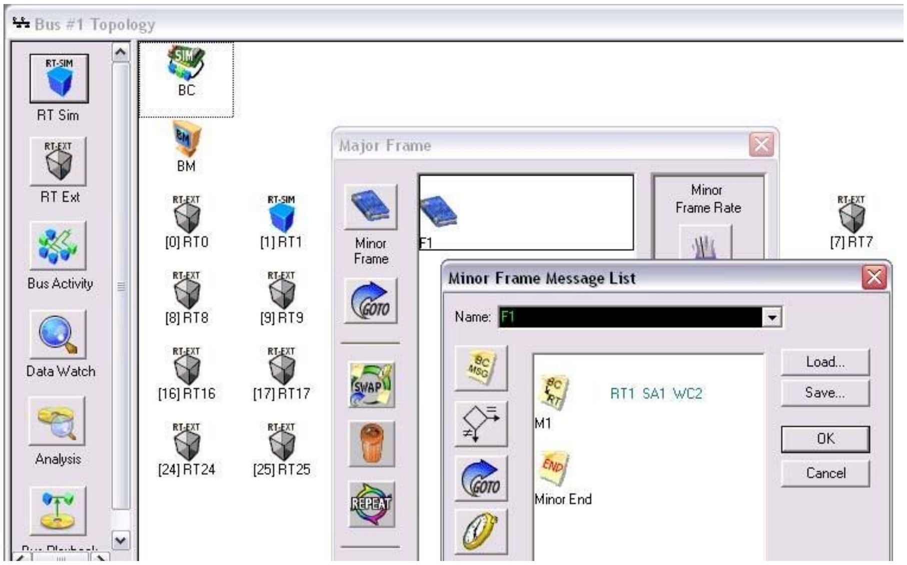


Error Injection from the Remote Terminal

Errors can be injected on STATUS words and (for transmit messages) on DATA words by the Remote Terminal.

For our examples, we will setup a topology with a BC and one RT (RT1). The BC will send the same message (RT1 RECEIVE SA1 WC2), with no error injection on the BC side.

We will demonstrate injecting PARITY errors from the RT for both RECEIVE and TRANSMIT messages. The other error types that can be injected from the RT are basically the same as for the BC.

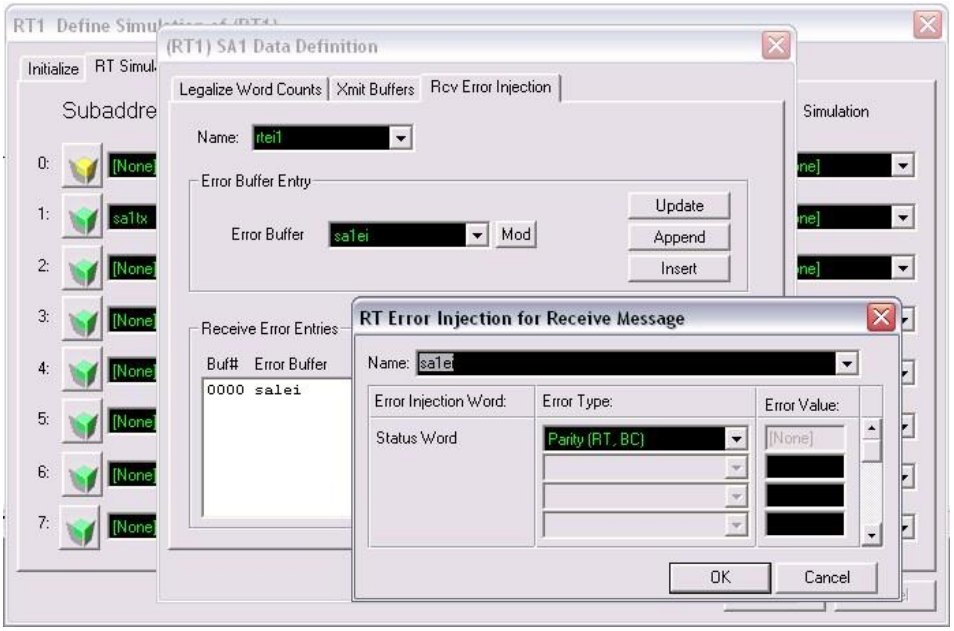


Name: F1

M1 RT1 SA1 WC2

Receive Message Errors

RT error injection on RECEIVE messages can only affect the STATUS word, since this is the only word that the RT sends in a receive message. Here we will inject a PARITY error on the receive status word.



Name: rtei1

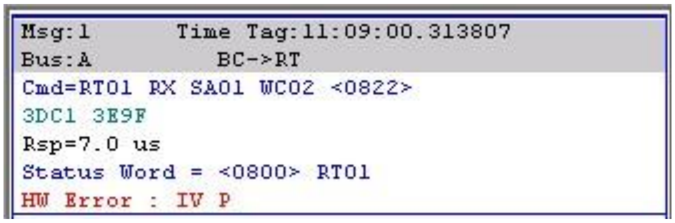
Error Buffer: sa1ei

Buf#	Error Buffer
0000	salei

RT Error Injection for Receive Message — Name: sa1ei

Error Injection Word:	Error Type:	Error Value:
Status Word	Parity (RT, BC)	[None]

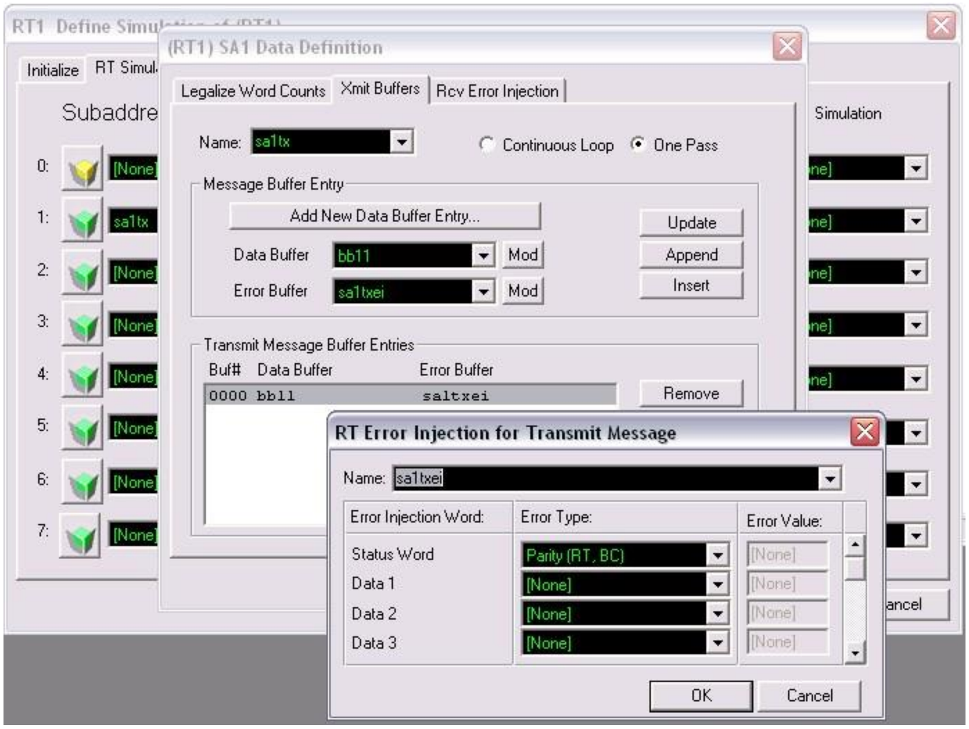
This is what the BM reports:


```
Msg:1        Time Tag:11:09:00.313807
Bus:A             BC->RT
Cmd=RT01 RX SA01 WC02 <0822>
3DC1 3E9F
Rsp=7.0 us
Status Word = <0800> RT01
HW Error : IV P
```

The Bus Monitor sees the parity error on the status word and also marks it as an invalid word.

Transmit Message Errors

RT error injection on TRANSMIT messages can affect STATUS or DATA words. We will now modify the BC message to send a RT-BC message. We also setup a PARITY error on the transmit status word.


RT1 Define Simulation of (RT1)
Initialize
RT Simul
Subaddre
Simulation
ancel
(RT1) SA1 Data Definition
Legalize Word Counts
Xmit Buffers
Rcv Error Injection
Message Buffer Entry
Add New Data Buffer Entry...
Update
Append
Insert
Mod
Mod
Transmit Message Buffer Entries
Remove
RT Error Injection for Transmit Message
OK
Cancel

Subaddresses: 0: [None]; 1: sa1tx; 2: [None]; 3: [None]; 4: [None]; 5: [None]; 6: [None]; 7: [None]

Name: sa1tx ○ Continuous Loop ● One Pass

Data Buffer: bb11

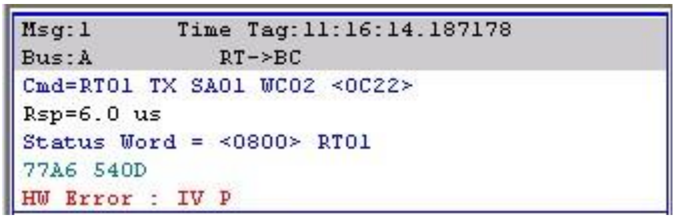
Error Buffer: sa1txei

Buf#	Data Buffer	Error Buffer
0000	bb11	saltxei

Name: sa1txei

Error Injection Word:	Error Type:	Error Value:
Status Word	Parity (RT, BC)	[None]
Data 1	[None]	[None]
Data 2	[None]	[None]
Data 3	[None]	[None]

Here is what the BM reports:



```
Msg:1       Time Tag:11:16:14.187178
Bus:A          RT->BC
Cmd=RT01 TX SA01 WC02 <0C22>
Rsp=6.0 us
Status Word = <0800> RT01
77A6 540D
HW Error : IV P
```

The Bus Monitor sees the parity error on the status word and also marks it as an invalid word.