File: lvds_tester.doc
Last updated: 22 March 2007

LVDS transmission tester based on the LTU board

Table of contents

Reference: [1] *Time domain compensation of cable induced distortions using passive filters for transmission of fast pulses*, G. Amsel et alt. The Review of Scientific Instruments, vol. 42, nb. 8, August 1971.
[2] *Evaluation of the twisted pair data transmission for the SCT*, R. Kowalewski et alt., ATLAS INDET-NO-172, 24 June 1997.
[3] *High-Density Digital Links*, Martin Mandl, PhD Theses, Technical University of Vienna, 9 February 2000.
[4] *SPD => PS 280 Mbits/s LVDS Link*, LHCb Calo Electronics Meeting, December 2003, Clermont- Ferrand.
[5] *LTU Preliminary Design Review*, revision 1.0, 15 October 2002, ALICE Trigger web site.
[6] *LTU Software Model*, revision 0.1, 23 August 2004, ALICE Trigger web site.
[7] *Random Trigger Generator with a Programmable Rate*, ALICE Trigger web site.
[8] *ADC Readout - Automatic Measurement of the Trigger Input Phase*, ALICE Trigger web site, CTP documentation, List of software notes, General - all boards.
[9] *Software LED - LTU board*, ALICE Trigger web site.
[10] *L0 Counters*, ALICE Trigger web site, CTP documentation, List of software notes, L0 board.
[11] *A multiplicity trigger based on the Time of Flight detector for the ALICE experiment*, A. Akindinov *et al*., 11th Workshop on Electronics for LHC and future Experiments, 12-16 September 2005, Heidelberg.

Introduction

Motivation

The LVDS format (*Low Voltage Differential Signalling*) has been selected for the transmission of the L0, L1 and L2 trigger inputs to the ALICE CTP, the L0 trigger outputs from the LTU board and the BUSY inputs from the sub-detectors. The main advantages of the format are its low power, low noise, high speed and a wide range of available driver and receiver ICs. But the LVDS standard sets the *maximum cable length* at *10m*, while most of the ALICE trigger system connections are in the range of 40m to 60m, some even as long as 100m. It has been demonstrated that - for a typical LHC rate of 40Mbit/s and with a good-quality cable - the transmission distortions can be successfully compensated with passive filters - *impedance equalizers* - making error-free LVDS-based transmission possible over distances of 100m and even longer [1][2][3][4].

The ALICE collaboration has specified and ordered - for most of the LVDS connections of the trigger system - a custom-developed cable, produced by Draka-Filéca company (some sub-detectors - TOF, HMPID, *etc*. - adopted a different solution [11]). The following are the main (measured) cable characteristics:

- single shielded twisted pair;
- outside diameter $\mathbf{d_o = 5.1mm}$;
- characteristic impedance $\mathbf{Z_0 = 124\Omega}$;
- propagation delay $\mathbf{t_{pd} = 4.5ns/m.}$

The passive impedance equalizer circuit has been developed (Fabio Formenti) and laboratory tests have verified the feasibility of LVDS transmission over the required length of cable, but more systematic and longer-term tests are still needed. For that purpose, the ALICE trigger LTU board (*Local Trigger Unit* [5]) has been "converted" into a rather sophisticated LVDS transmission tester.

The description of the board operation and the list of available measurement options are presented in the following sections.

Implementation

It just happens that the LTU board [5], designed for a very different application, contains all the necessary "ingredients" for an LVDS tester capable of measuring the transmission bit-error rate:

- the LVDS drivers and receivers, connected to the front panel connectors;
- 40MHz clock (BC) and a programmable clock delay line with 31 steps of 1ns;
- 8-bit ADC that can measure the phase of input signals in respect to the BC clock;
- a fast FPGA with high capacity (ALTERA Cyclone EP1C12F324C-7) - for the implementation of the instrument's logic;
- VME interface - to enable a processor to control and monitor the measurement procedure.

As a result, the "conversion" of a *normal LTU* into an *LVDS tester* consists only of loading (temporarily) a different firmware configuration (*lvds_tester*) into the board's FPGA.

The following LTU VME control/monitoring words are "reused" in the *LVDS Tester* implementation; their format and their function has remained identical to their original LTU application and are fully documented in [6]:

Mnemonic	*Address*
VERSION_ADD	H"20"
TEST_ADD	H"30"
BC_STATUS	H"31"
BC_DELAY_ADD	H"132"
ADC_START	H"33"
ADC_DATA	H"34"
SCOPE_SELECT	H"154"
SOFT_LED	H"57"
SSM_COMMAND	H"67"
SSM_START	H"68"
SSM_STOP	H"69"
SSM_ADDRESS	H"6A"
SSM_DATA	H"6B"
SSM_STATUS	H"6C"
PLL_RESET	H"6F"
COPY_COUNT	H"75"
COPY_BUSY	H"76"
COPY_CLEARADD	H"77"
COPY_READ	H"78"
CLEAR_COUNTER	H"16B"

The *LVDS Tester*-specific VME words - all shown in **Figure 1** - are explained in the following sections.

Block diagram

The block diagram of the *LVDS Tester* is shown in **Figure 1**; the *signal names* are printed red; the *control and/or monitoring words* that affect the operation of a logic block are printed blue, adjacent to the block symbol.

Inputs and outputs

The *Pattern* output is simultaneously connected to three L0 connectors (front panel, **Figure 2**); each output is driven by the SN65LVDM31D differential driver (Texas Instruments).

The instrument provides two identical channels; one could be used for measurements of a cable that is being tested; an already tested cable - short, no transmission problems - could be simultaneously connected to the other channel and used as a measurement reference.

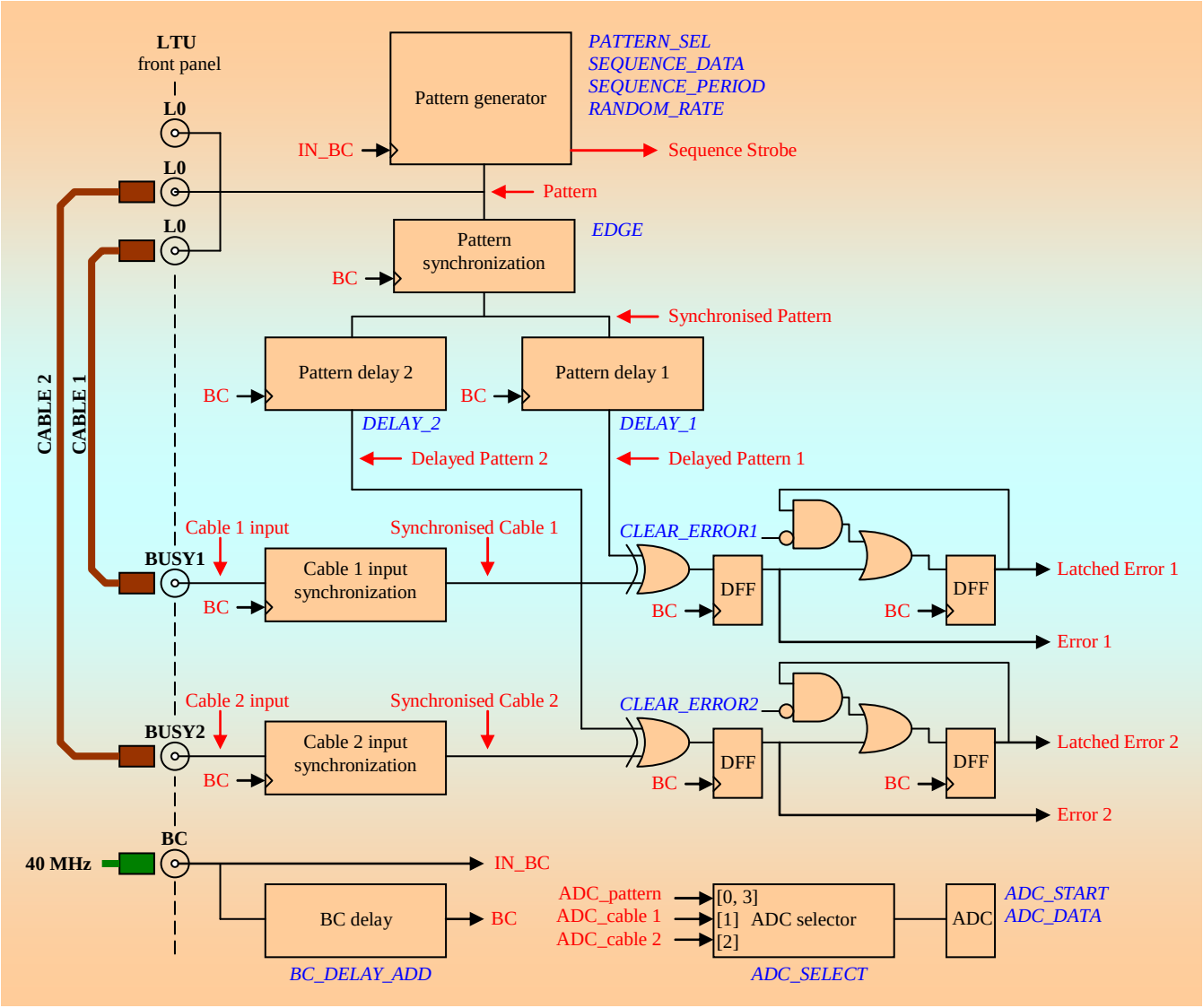


Figure 1 Block diagram of the *LVDS Tester*

The cables are connected to the front panel BUSY1/BUSY2 connectors (**Figure 2**); the line receiver is the SN65LVDS32BD IC (Texas Instruments); the line terminating resistor has a value of 100Ω.

Pattern generator block

The block generates three output patterns; the pattern selection is done with the *Pattern Selector* word:

- the s*equence* is a 24-bit programmable data stream, transmitted repeatedly at a programmable rate; the data content is set with the *Sequence Data* word; the *Sequence* period is defined with the *Sequence Period* word; the signal *Sequence Strobe* (**Figure 1**) coincides in time with the first bit of the data stream - it is used only internally for counting and scope synchronisation.
- the programmable rate of the *random* signal is set by the *Random Rate* word;
- the *toggling* output - the BC clock divided by two - is required for phase measurements.

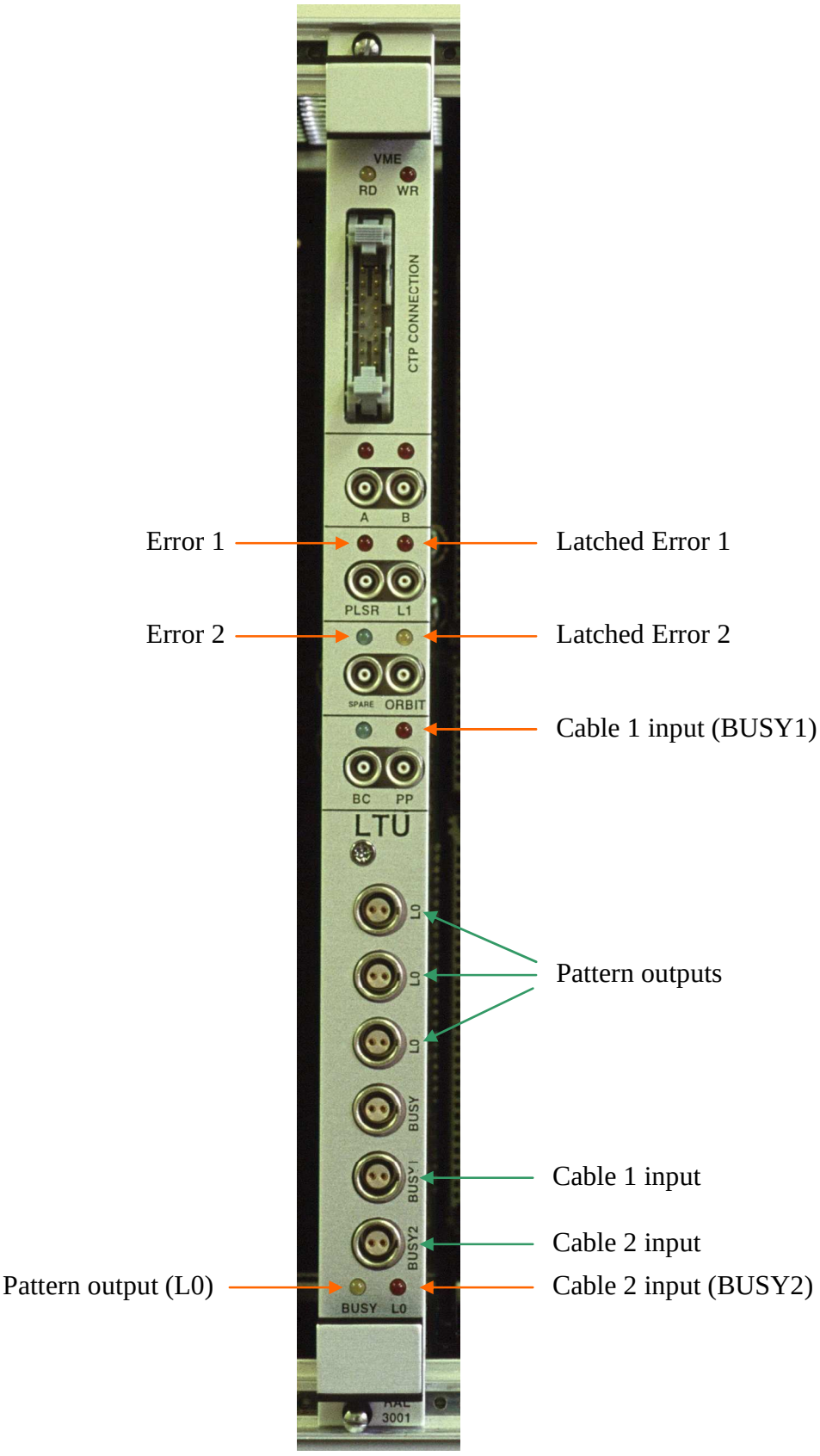


Figure 2 *LVDS Tester* front panel LEDs and the cable connections

Pattern Select word

Pattern Select word (*PATTERN_SEL*, address H”121”, write and read):

Data bit	Selection
D[1..0] = 0	Output cleared - no option selected
D[1..0] = 1	24-bit programmable *sequence*
D[1..0] = 2	*Random pattern* with a programmable rate
D[1..0] = 3	*Toggling output* (for synchronisation)

Sequence Data word

Sequence Data word (*SEQUENCE_DATA*, address H”123”, write and read):

Data bit	Signal
D[23..0]	24-bit data stream pattern; bit D[23] transmitted first

Sequence Period word

Sequence Period word (*SEQUENCE_PERIOD*, address H”122”, write and read):

Data bit	Signal
D[11..0]	*Sequence* period - in BC intervals (25ns).

The available range is 0 to 102.4µs; setting of the period below 25 shall corrupt the data stream.

Random pattern

The circuit used to generate the *rundom trigger* and the properties of the pattern are described in [7].

The pulse distribution is pseudo-random - the pattern repeats itself every ~53s ($[2^{31}-1]$ BC intervals). The average rate of the random trigger is defined by the control word *Random Rate.*

Random Rate word

Random Rate word (*RANDOM_RATE*, address H”124”, write and read):

Data bit	Signal
D[30..0]	Average rate of the *random* pattern

The content of the 31-bit register is a number of pulses generated within each pseudo-random pattern repetition period (range: 1 to $[2^{31}-1]$).

Example: For D[30..0] = H"3FFFFFFF", the average rate is 50% - equal number of 0 and 1 states; for D[30..0] = H"7FFFFFFF", the output is permanently on - 100% rate.

Synchronization

The 40 MHz BC input of the LTU board, *IN_BC*, is only used to clock the *Pattern Generator* block (**Figure 1**); the rest of the *LVDS Tester* logic is synchronised with the *BC* clock, generated by the FPGA's phase locked loop that uses as a reference the delayed *IN_BC* signal. The clock delay is programmable - 31 steps of 1ns each; the control VME word *BC_DELAY_ADD* is explaind in [6], section **3.7.2**.

The three inputs to the main, *BC* clocked logic domain - *Pattern*, *Cable 1* and *Cable 2* (**Figure 1**) - have to be properly synchronised in order to avoid - or ,at least, understand and anticipate - possible violations of the *set-up time* (t_{su}) and the *hold time* (t_h) rules. The procedure requires the measurement of the signal phase in respect to the current setting of the *BC* clock (*BC* delay). The ADC on the LTU board is used to measure the phase - the procedure and the use of the VME control words *ADC_START* and *ADC_STOP* are explained in [8]. Selection of a signal for the phase measurement is done with the VME control word *ADC Select*.

The pattern used during the phase measurements is the *toggling* output.

ADC Select word

ADC Select word (*ADC_SELECT*, address H"120", write and read):

Data bit	Selection
D[1..0] = 0,3	*Pattern* signal - *ADC_pattern* input (**Figure 3**)
D[1..0] = 1	*Cable 1* signal - *ADC_cable1* input (**Figure 4**)
D[1..0] = 2	*Cable 2* signal - *ADC_cable2* input (**Figure 4**)

Pattern signal synchronization

The synchronization circuit for the *Pattern* signal is shown in **Figure 3**. The VME control word *Synchronization Edge* selects the sampling mode. When the *Pattern* is a *toggling* output, the *ADC_pattern* signal is the input for the phase measurement.

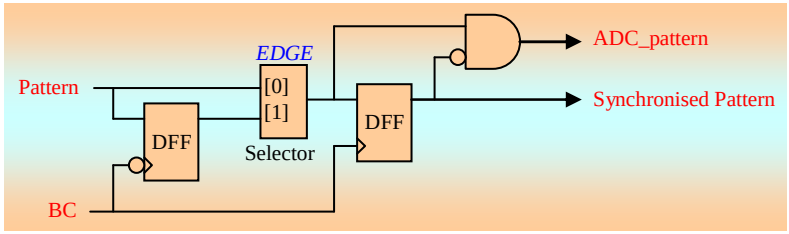


Figure 3 Synchronisation of the *Pattern* signal

Synchronization Edge word

Synchronisation Edge word (*SYNCH_EDGE*, address H"125", write and read):

Data bit	Selection
D[0] = 0	*Pattern* signal is clocked with the positive *BC* edge
D[0] = 1	*Pattern* signal is first sampled with the negative *BC* edge and then re-clocked with the positive edge (**Figure 3**)

Cable 1(2) input synchronization

The synchronization circuit for the *Cable 1(2)* input is shown in **Figure 4**. When the *Pattern* is a *toggling* output, the *ADC_cable1(2)* signal is used as an input for the phase measurement.

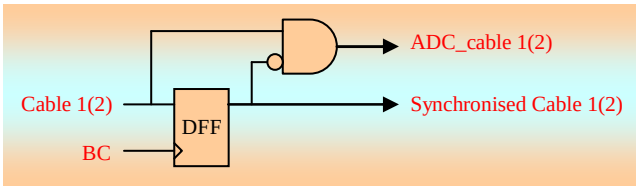


Figure 4 Synchronisation of the *Cable 1(2)* signal

Pattern delay

In order to evaluate the quality of the *Pattern* transmission, the *Synchronised Pattern* signal (**Figure 1**) needs to be aligned in time with the *Synchronised Cable 1(2)* input. The alignment is achieved by delaying the *Synchronised Pattern* for an appropriate number of BC intervals (25ns). The delay is set with the VME words *Delay 1* and *Delay 2*. The alignament procedure is described in the **Measurements** section.

Delay 1(2) word

Delay 1 word (*DELAY_1*, address H"126", write and read),
Delay 2 word (*DELAY_2*, address H"127", write and read),

Data bit	Signal
D[4..0]	Delay of the *Delayed Pattern 1(2)* signal in BC intervals (25ns)

The delay range - 0 to 775ns - is sufficient for cable length of up to ~155m (assuming the propagation delay of 5ns/m).

Error checking logic

The error checking logic is depected in **Figure 1**. In each *BC* clock interval, the transmitted pattern - *Delayed Pattern 1(2)* signal - is compared with the received pattern - *Synchronised Cable 1(2)* signal. If their states - high or low - are not the same, the corresponding *Error 1(2)* signal is asserted. The *Lached Error 1(2)* signal is set by the first asserted *Error 1(2)* signal and remains so until it is cleared by the VME control word *Clear Error 1(2)* (*CLEAR_ERROR1*, address H"128", *CLEAR_ERROR2*, address H"129"; write only, dummy data). The current status of the *Lached Error 1(2)* signal can be read with the VME word *Error Status*.

Error Status word

Error Status word (*ERROR_STATUS*, address H"80", read only):

Data bit	Signal
D[0]	*Latched Error 1* signal
D[1]	*Latched Error 2* signal

Front panel LEDs

When the LTU is configured as the *LVDS Tester*, some of the existing front panel LEDs are used to display the status of internal tester signals. The new allocation is shown in **Figure 2**. The LEDs that are not re-allocated retain their original meaning ([6], section **3.3.1**).

SoftLED option

The *SoftLED* option in the *LVDS Tester* configuration is identical to the option available on the original LTU board. The option is described in [9].

Software LED word

Software LED read word (*SOFT_LED*, address H"57"):

Data bit	Signal
D[0]	*LED[1] - VME read strobe*
D[1]	*LED[2] - VME write strobe*
D[2]	*LED[3] - SCOPE_A* signal
D[3]	*LED[4] - SCOPE_B* signal

SCOPE_A/B signal is the signal selected by the *SCOPE_SELECT* word (address H"54").

SnapShot option inputs

The use of the *SnapShot* option in the *LVDS Tester* configuration is the same as on the original LTU board. The bit allocation is given in **Table 1**.

Data bit	***Signal mnemonic***
0	*Synchronised Pattern*
1	*Delayed Pattern 1*
2	*Delayed Pattern 2*
3	*Synchronised Cable 1*
4	*Synchronised Cable 2*
5	*Error 1*
6	*Latched Error 1*
7	*Error 2*
8	*Latched Error 2*
9 - 16	GND (*not used*)
17	*VME access (Strobe)*

Table 1 Bit allocation of the *SnapShot* memory for the *LVDS Tester*

ScopeProbe option inputs

Signal selection for the *ScopeProbe* option is the same as on the original LTU board (*SCOPE_SELECT* word - [6], section **3.18**). The following is the list of signal input allocations, the same for both groups A and B:

Selection code	Signal
0	*BC*
1	*IN_BC*
2	*Pattern*
3	*Synchronised Pattern*
4	*Delayed Pattern 1*
5	*Delayed Pattern 2*
6	*ADC_cable 1*
7	*ADC_cable 2*
8	*ADC_pattern*
9	*Cable 1* input
10	*Cable 2* input
11	*Error 1*
12	*Error 2*
13	*Latched Error 1*
14	*Latched Error 2*
15	*Synchronised Cable 1*
16	*Synchronised Cable 2*
17	*ADC_in*
18	*Sequence Strobe*
19 - 28	*GND*
29	*VME read*
30	*VME write*
31	*VME access (read or write)*

LVDS Tester counters

There are **7** LVDS Tester counters; they all have a capacity of 32 bits. The counters are divided in two groups:

- *Time counters* - timers (3 off),
- *Fast-signal counters* (4 off),

A general description of the counters is given in [10]; the text also includes the definition of the VME interface and outlines the software sequence that is used to read the counters.

The **Table 2** and **Table 3** list the counters of the *LVDS Tester* configuration; the tables include the counter addresses in the *CounterCopy* memory.

Time counters (**Table 2**)

Address	*Signal mnemonic*	*Signal description*
0	VCC	Elapsed time
1	!(*Latched Error 1*)	Time until first *Error 1*
2	!(*Latched Error 2*)	Time until first *Error 2*

Table 2 LVDS Tester *time* counters - *CounterCopy* memory allocation

"Elapsed time" timer (address 0) counts continuously; it is intended for real time measurements.

Fast-signal counters (**Table 3**)

Address	*Signal mnemonic*
8	*Error 1*
9	*Error 2*
10	*Synchronised Pattern*
11	*Sequence Strobe*

Table 3 LVDS Tester *fast-signal* counters - *CounterCopy* memory allocation

Measurements

The measurements are presented in the order in which they should be performed. The **Tp**, the **Tc** and the **D** measurements are preparatory - they provide the necessary parameters to be used for the "real" measurements that evaluate the quality of the LVDS link.

Tp - transition of the *Synchronised Pattern* signal

Tp is a BC delay at which the transition of the *Synchronised Pattern* signal occurs.

- Select the *toggling* pattern -> *PATTERN_SEL* D[1..0] = 3.
- Select synchronization with a *positive BC edge* -> *SYNCH_EDGE* D[0] = 0.
- Select the *ADC_pattern* signal as the ADC input -> *ADC_SELECT* D[1..0]=0.
- Using the ADC, measure the phase shift of the toggling *Pattern* (clocked by the *IN_BC*) in respect to the *BC* clock, for the whole BC delay range (BC_DELAY_ADD D[4..0] = 0 to 31).
- Plot the results of the scan - ADC readings (0 to 255) versus the BC delay (0 to 31); repeat the ADC measurement at each point (10 times, for example) in order to check the consistency.
- The **Tp** is the BC delay at which the transition of the ADC reading occurs.

The expected value is around 16 - 18; for the same LTU board and the same version of the *LTU Tester* firmware - it should remain unchanged.

A sample plot is shown in **Figure 5**.

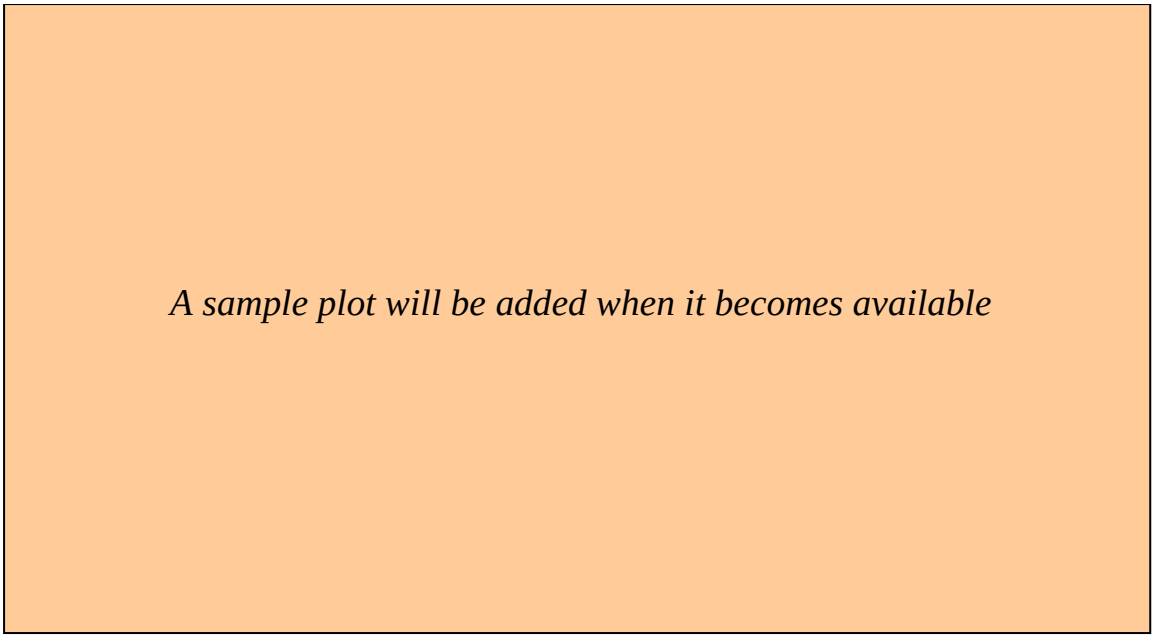


Figure 5 Measurement of the phase of the *Pattern* signal

Selection of the *BC* clock edge for the *Synchronised Pattern* signal

For all the remaining measurements - all the measurements except for the **Tp** transition scan - the selection of the *BC* clock edge for the *Synchronised Pattern* signal should be done in accordance with the following *Edge Rule*:

> For the BC delay inside the interval [(**Tp** - 3), (**Tp** + 3)] use the *negative* edge (*SYNCH_EDGE* D[0] = 1); for the delays outside the interval, use the *positive* edge (*SYNCH_EDGE* D[0] = 0).

The *Edge Rule* is presented graphically in **Figure 7**.

Tc - transition of the *Synchronised Cable* signal

Tc is a BC delay at which the transition of the *Synchronised Cable* signal occurs.

- Select the *toggling* pattern -> *PATTERN_SEL* D[1..0] = 3.
- In this measurement, the selection of the *BC edge* (*SYNCH_EDGE*) is irrelevant.
- Select the *ADC_cable* signal as the ADC input -> *ADC_SELECT* D[1..0] = 1 - for *Cable 1*/ 2 - for *Cable 2*.
- Using the ADC, measure the phase shift of the toggling *Cable 1/2* input in respect to the *BC* clock, for the entire BC delay range (*BC_DELAY_ADD* D[4..0] = 0 to 31).
- Plot the results of the scan - ADC readings (0 to 255) versus the BC delay (0 to 31); repeat the ADC measurement at each point (10 times, for example) in order to check the consistency. A sample plot is shown in **Figure 6.**
- The **Tc** is the *BC* delay at which the transition of the ADC reading occurs (depending on the cable length, there might be two transitions - **Tc** and **Tch**, **Figure 7**).

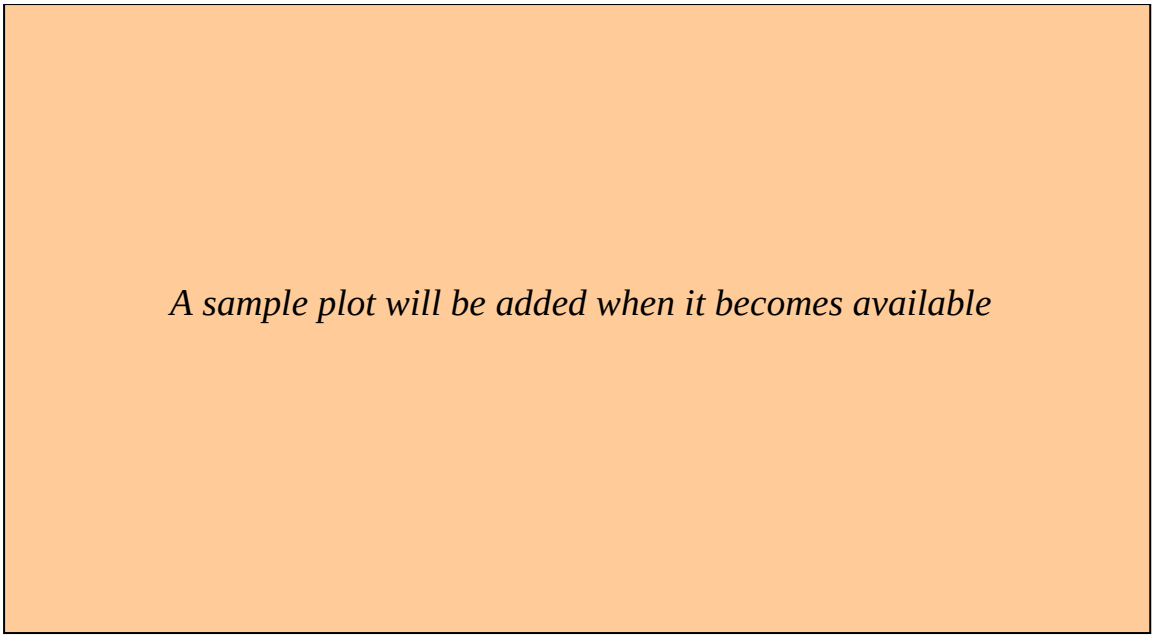


Figure 6 Measurement of the phase of the *Cable* signal

D - *Pattern* delay for a given cable length

D is a *Pattern* delay that ensures that the original pattern and the version received *via* the cable coincide in time at the point where the data verification is performed (**Figure 1**); the delay is dependant upon the length of the used cable.

- Select the *sequence* pattern -> *PATTERN_SEL* D[1..0] = 1.
- Select the *sequence* with all 24 bits asserted -> *SEQUENCE_DATA* D[23..0] = H"FFFFFF"; the sequence is likely to return the least number of errors in case of a cable of poor quality, or too long a cable.
- Select the *sequence* period of ~1.5µs (60 BC intervals) -> *SEQUENCE_PERIOD* D[11..0] = 60; the period is long enough to avoid the sequence overlapping for cable length of up to 160m.
- Calculate the value of the **Ts** following the rules presented in **Figure 7**; the formula to be used - **Ts** = **Tc**+12, or **Ts** = **Tc**-12 - depends upon the shape of the **Tc** plot; the offset of 12 provides for sampling at the centre of the *Cable* signal, the value that is likely to return the least number of errors.
- Select **Ts** as the BC delay -> BC_DELAY_ADD D[4..0] = **Ts**.
- For the BC delay of **Ts**, apply the *Edge Rule* to select the *BC edge* (*SYNCH_EDGE* word).
- Using the *Error* signal counter, find the number of errors for the entire *Pattern* delay range (*DELAY_1/2* D[4..0] = 0 to 31); for each measurement, count the errors over a large number of sequences (1000, for example) and then normalise the number of errors on a *per sequence* basis (use the *Sequence Strobe* counter or the *Elapsed Time* timer).
- Plot the results of the scan - normalised number of errors versus the *Pattern* delay (0 to 31).
- The **D** is the only *Pattern* delay at which the number of errors is zero; in case of a cable of poor quality, or too long a cable, there might not be a delay that returns no errors - select instead the point with the smallest number of errors.

Measurement of the Sampling *Window*

The measurement of the *Sampling Window* uses the values of parameters **Tp**, **Tc**,**Ts** and **D** found in the preparatory measurements.

- Select the *random* pattern -> *PATTERN_SEL* D[1..0] = 2.
- Set the average *rate* of the *random pattern* at 50% -> *RANDOM_RATE* D[30..0] = H"3FFFFFFF".

  Note 1: It might happen that the signal transmission with a much lower rate (1%, for example), or with a much higher rate (99%) would register more errors - the case of individual, positive or negative pulses separated by long pauses. The same measurements could also be done with a repeated *sequence* of a fixed configuration.

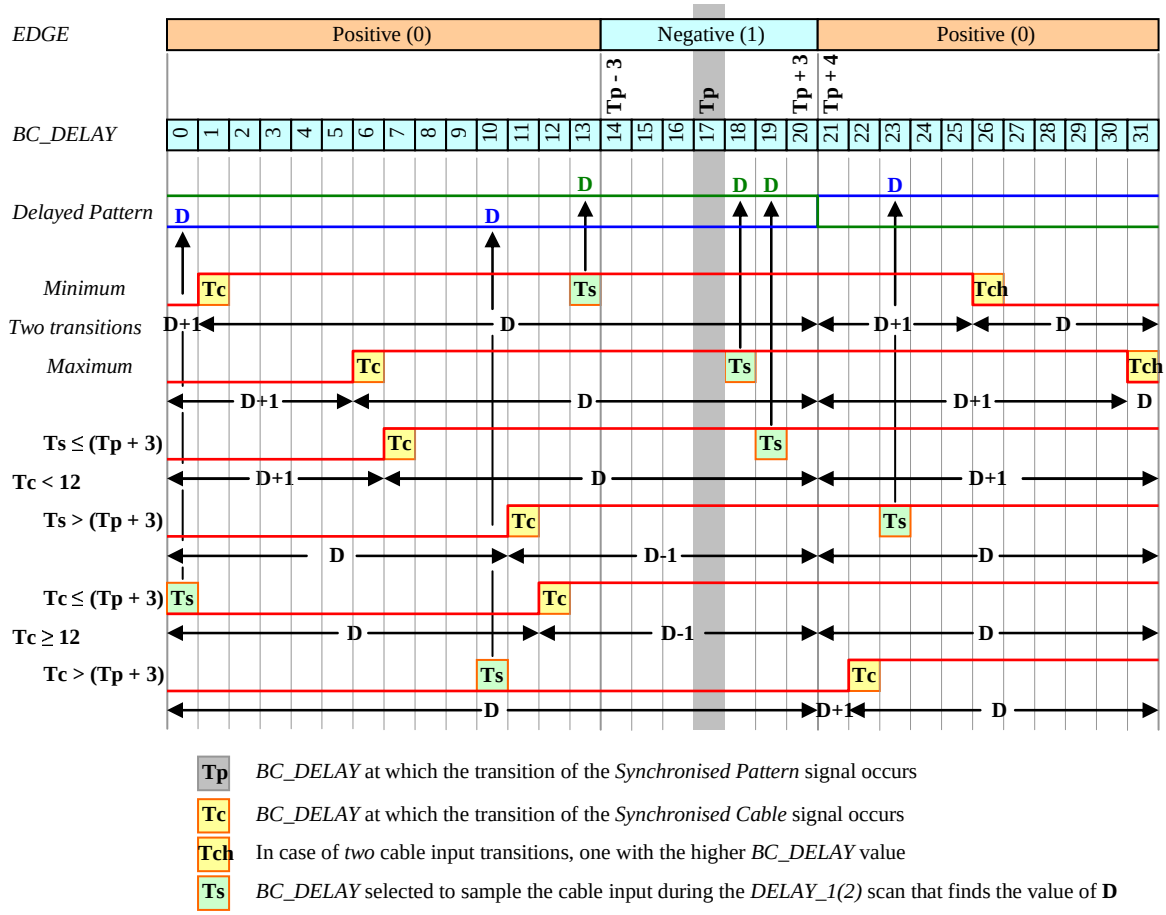
- Using the *Error* signal counter, find the number of errors for the entire *BC* delay range (*BC_DELAY_ADD* = 0 to 31); for each *BC* delay setting, apply the *Edge Rule* to select the *BC edge* (*SYNCH_EDGE* word); for each *BC* delay setting, set the *Pattern* delay (*DELAY_1/2* D[4..0]) to **D, D+**1, or **D-**1, according to the rules presented in **Figure 7**.

  Note 2: The pseudo-random pattern repeats itself every ~53s and, in order to test all the data patterns, the error counting should be done during a time interval of that length (or a multiple of it). With the clock frequency of 40MHz, there are ~ $2 \cdot 10^9$ bit transfers during the 53s interval; if N errors were counted, the corresponding bit error rate would be ~$N/2 \cdot 10^{-9}$. Note that it would take approximately half an hour to complete the measurement.

  If, instead of a *random* patter, characteristic *sequences* were used (example: alternating 1s and 0s - 24 of them - followed by a longish pause - 50µs or so), the number of trabsmitted *sequences*, repeated for each *BC* delay setting, should be sufficient to provide a meaningful statistics.

- Plot the results of the scan - the number of errors (logarithmic scale) versus the *Pattern* delay (0 to 31). A sample plot is shown in **Figure 8**.

The width of the *Sampling Window* is the number of consecutive *BC* delay bins with no error recorded. For a good quality connection, the window should be 18 or higher.



Tp *BC_DELAY* at which the transition of the *Synchronised Pattern* signal occurs

Tc *BC_DELAY* at which the transition of the *Synchronised Cable* signal occurs

Tch In case of *two* cable input transitions, one with the higher *BC_DELAY* value

Ts *BC_DELAY* selected to sample the cable input during the *DELAY_1(2)* scan that finds the value of **D**

Selection of the **Ts** and settings of the *DELAY_1(2)* during the *BC_DELAY* scan [0, 31]

1. Two cable input transitions - **Tc** and **Tch:** **Ts = Tc + 12**

BC_DELAY	[0, (Tc-1)]	[Tc, (Tp+3)]	[(Tp+4), (Tch-1)]	[Tch, 31]
DELAY_1(2)	D+1	D	D+1	D

2. Single transition, **Tc < 12:** **Ts = Tc + 12**

2a. Ts ≤ Tp + 3

BC_DELAY	[0, (Tc-1)]	[Tc, (Tp+3)]	[(Tp+4), 31]
DELAY_1(2)	D+1	D	D+1

2b. Ts > Tp + 3

BC_DELAY	[0, (Tc-1)]	[Tc, (Tp+3)]	[(Tp+4), 31]
DELAY_1(2)	D	D-1	D

3. Single transition, **Tc ≥ 12:** **Ts = Tc - 12**

3a. Tc ≤ Tp + 3

BC_DELAY	[0, (Tc-1)]	[Tc, (Tp+3)]	[(Tp+4), 31]
DELAY_1(2)	D	D-1	D

3b. Tc > Tp + 3

BC_DELAY	[0, (Tp+4)]	[(Tp+4), (Tc-1)]	[Tc, 31]
DELAY_1(2)	D	D+1	D

Figure 7 Selection of the **Ts** and setting of the *Pattern* delay

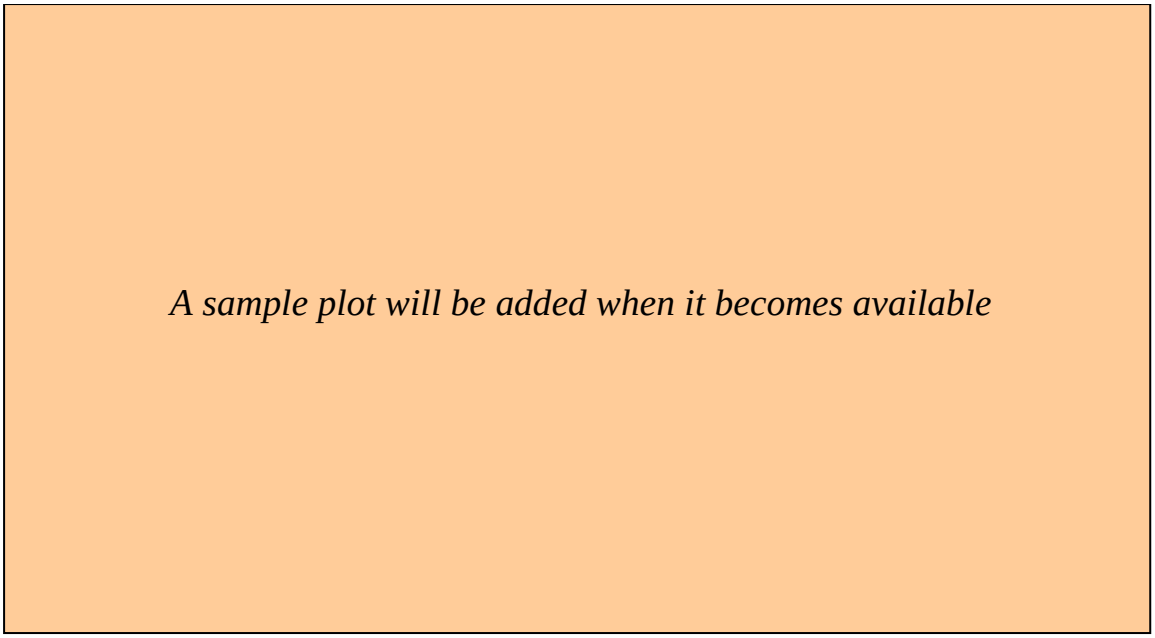


Figure 8 Measurement of the *Sampling Window*

Bit Error Rate (BER) measurement

Measurement of the width of the *Sampling Window* provides the BER value over approximately 10^9 bit transmissions. According to an accepted *rule of thumb*, if no error is detected at that level, the data link under test is considered reliable. Nevertheless, one is likely to want to check the operation of the link at a set sampling point (*BC* delay) and for a longer period of time - a number of hours, overnight, *etc*. . Sampling at the **Ts** should give the best result, but the point of real importance is at **Tc** = ± 5ns - somewhat worth than the worst working margin (**Tc** = ± 6.25ns) left after the CTP trigger input synchronization procedure.

The BER measurement uses the values of parameters **Tp**, **Tc**,**Ts** and **D** found in the preparatory measurements.

- Select the *random* pattern -> *PATTERN_SEL* D[1..0] = 2.
- Set the average *rate* of the *random pattern* at 50% -> *RANDOM_RATE* D[30..0] = H"3FFFFFFF".

  It might happen that the signal transmission with a much lower rate (1%, for example), or with a much higher rate (99%) would register more errors. The same measurements could also be done with a repeated *sequence* of a fixed configuration - see *Note 1* and *Note 2*.

- Set the *BC* delay for the sampling point of interest - *BC_DELAY_ADD* = **Tc** ± 5ns, for example (if both points are possible, select the lower one).
- Apply the *Edge Rule* to select the *BC edge* (*SYNCH_EDGE* word).
- For the adopted *BC* delay, set the *Pattern* delay (*DELAY_1/2* D[4..0]) to **D,** **D+**1, or **D-**1, according to the rules presented in **Figure 7**.

- Using the *Error* signal counter, find the number of errors during a period timed with the *Elapsed Time* counter (in units of 0.4μs); as an example, a period of ~7 hours (overnight run) is required to evaluate the BER at the 10^{12} level.
- Read the counters in regular interval - every minute, or so - and plot the historygram of the error count (logarithmic scale); the plot will show whether the errors occurred in a burst or they were spread evenly over the entire period of measurement. A sample plot is shown in **Figure 9**.

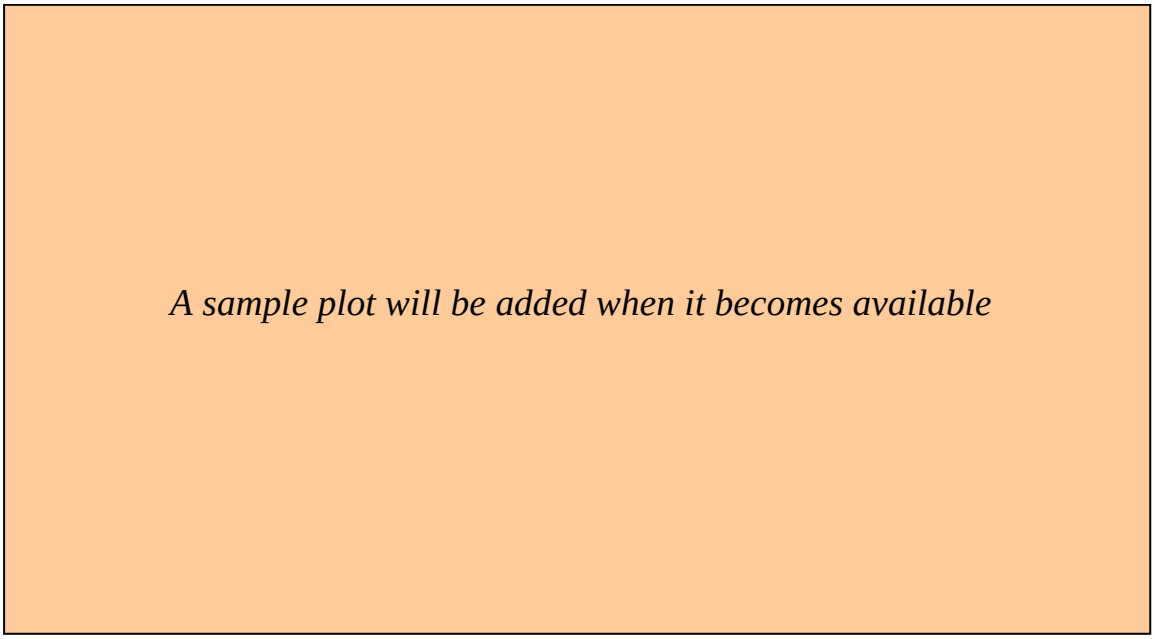


Figure 9 Historygram of the BER