Agilent 81250 Parallel Bit Error Ratio Tester

System User Guide

Agilent Technologies

Important Notice

Manual Part Number

5988-4887EN

Revision

Revision 5.1a, May 2003

Printed in Germany

Agilent Technologies
Herrenberger Straße 130
D-71034 Böblingen
Germany

Authors: t3 medien GmbH

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING/DANGER

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Trademarks

Windows NT ® and MS Windows ® are U.S. registered trademarks of Microsoft Corporation.

Contents

About this Guide

This user guide provides comprehensive information on the hardware and standard user software of the Agilent 81250 Parallel Bit Error Ratio Tester.

This guide does not cover:

- System installation/update
- System configurations
- Technical specifications
- The Agilent 81250 ParBERT Measurement Software

Information on these topics is available in dedicated manuals.

E-mails

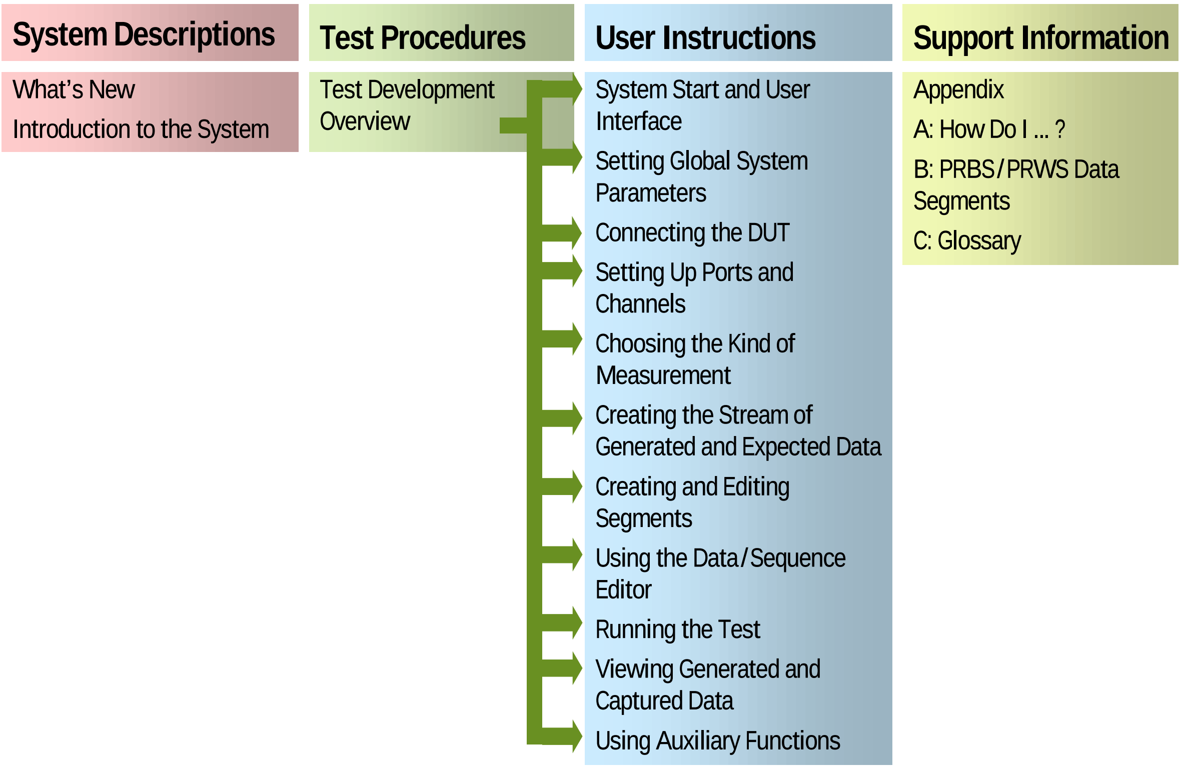
You can contact the ParBERT support under the e-mail address parbert_support@agilent.com.

Structure of the System User Guide

As the ParBERT System User Guide comprises quite many chapters, you may wish to have some kind of overview to decide how to make optimum use of this document. This first chapter provides that overview.

General Structure

Actually, the information is organized into four sections, though these sections are not explicitly labeled in the manual:



System Descriptions

What’s New
Introduction to the System

Test Procedures

Test Development Overview

User Instructions

System Start and User Interface
Setting Global System Parameters
Connecting the DUT
Setting Up Ports and Channels
Choosing the Kind of Measurement
Creating the Stream of Generated and Expected Data
Creating and Editing Segments
Using the Data/Sequence Editor
Running the Test
Viewing Generated and Captured Data
Using Auxiliary Functions

Support Information

Appendix
A: How Do I ... ?
B: PRBS/PRWS Data Segments
C: Glossary

Figure 1 User Guide Structure

The chapter *Test Development Overview* describes the sequence of steps to be performed when developing, executing, and verifying a device test. It points to the corresponding instructions.

Contents of the Chapters

What's New?

If you have already been working with the Agilent 81250 Parallel Bit Error Ratio Tester and received a new software release, you should read this section. It informs you about the changes that have been made and the enhancements that have been added.

Introduction to the System

Newcomers should read this chapter in order to find out what they can expect from the Agilent 81250 Parallel Bit Error Ratio Tester, what the hardware components do, and how the system works.

Advanced users will use this chapter as a reference for the terms that appear on the user interface.

Test Development Overview

This chapter summarizes the steps which are necessary for testing a device with the Agilent 81250 Parallel Bit Error Ratio Tester—from setup to results. As the system has been designed for ease of use, these steps can be performed by clicking tool bar icons.

System Start and User Interface

You can of course start the system and find out yourself how the user interface works and how it is operated. The online help will assist you. But you will probably miss a couple of details which could make your life easier. Such details are discussed in this chapter.

Setting Global System Parameters

This and the following chapters provide task-related information. They are sorted according to the step-by-step procedures stated in the *Test Development Overview* chapter.

Global system parameters refer to the setup of the clock module.

Setting the test frequency is easy. But if you wish to take advantage of the system's capabilities, you will also need some general information about the system's timing principles from the *Introduction* chapter.

Connecting the DUT

This chapter explains how to use the Connection Editor. The Connection Editor is the first window after starting the Agilent 81250 Parallel Bit Error Ratio Tester User Software.

Setting Up Ports and Channels

This chapter explains how to specify the timing, voltages, operating modes and more of the generator and analyzer frontends.

Choosing the Kind of Measurement

The Agilent 81250 Parallel Bit Error Ratio Tester User Software allows to capture data, to compare received data in real time with expected data, and to measure the bit error rate. This chapter explains the details.

Creating the Stream of Generated and Expected Data

Testing a device most often requires two data streams—one that is generated, and one that is expected and compared with the DUT output. This chapter explains how these streams can be set up with the Sequence Editor.

To understand this and the following chapters, you will need some basic information about the system’s data generation principles from the *Introduction* chapter.

Creating and Editing Segments

The data to be sent to or expected from the DUT is organized as data segments. This chapter describes how to create such segments.

Using the Data / Sequence Editor

The Data/Sequence Editor is a very versatile tool which is described in this chapter.

Running the Test

This chapter explains how a test is started and stopped.

Viewing Generated and Captured Data

Depending on the kind of measurement, the ParBERT user software offers several kinds of result displays. This chapter explains how to enable and operate these displays.

Using Auxiliary Functions

Auxiliary functions like delay compensation, data import/export, or the Command Line Editor are described in this chapter.

Appendix A: How Do I ... ?

This appendix provides answers to frequently asked questions. If you need to perform a special test or need a special function or mode of operation, you should consult this chapter before calling the Agilent support.

Appendix B: PRBS / PRWS Data Segments

This appendix explains the creation and properties of the pseudo random bit/word streams generated or expected by the Agilent 81250 Parallel Bit Error Ratio Tester.

Appendix C: Glossary

All the specific terms used by the user interface are explained in this guide, most of them in the *Introduction* chapter. The glossary in the appendix provides a quick reference.

Index

The alphabetical index register has been prepared to assist you in finding any bit of information as simply and quickly as possible.

About the ParBERT Measurement Software

The Measurement Software for the Agilent 81250 Parallel Bit Error Ratio Tester extends the capabilities of the standard system. It provides additional and unequaled features for R&D and production with regard to both test time and precision.

Measurement user interface

The Agilent 81250 Measurement Software has its own user interface, independent from the user interface of the standard user software.

The user interface of the Agilent 81250 Measurement Software consists of a workspace (a window frame) that holds the windows of one or a number of measurements.

Setting

Before you can use the Agilent 81250 Measurement Software you have to create and save a **setting** with the Agilent 81250 User Software.

A setting contains all the system setup information, including the pins of the device to be tested, their connections to the system, the test frequency, the clock source, the timings and levels to be used, the data to be sent or expected, and so on. Once a suitable setting has been stored, the Agilent 81250 User Software does not have to be active for performing the measurements.

Documentation

The Agilent 81250 Measurement Software comes with its own manuals.

There is a general guide for all measurements and there are single guides for the individual measurements.

The general guide is the *Agilent 81250 ParBERT Measurement Software Framework User Guide*. The guides for the measurements are named according to the measurements.

About Special ParBERT Applications

The ParBERT user software provides dedicated tools (utility programs) and manuals for special applications. This is part of Agilent’s ongoing effort to offer not only general purpose instruments but also turnkey solutions.

When you receive a new revision of the ParBERT software, it is recommended that you inspect the updated system documentation coming with that revision.

Here are some examples:

Testing SFI-5 devices

SFI-5 (SERDES Framer Interface 5) specifies the interface between the framer and the serializer/deserializer used in transceivers for synchronous optical networks (SONET). For testing such devices, ParBERT provides special tools and the manual *Testing SFI-5 Devices*.

Generating SONET / SDH frames

ParBERT provides also a tool for generating test data that is formatted according to the SONET and/or SDH specifications. The related document is the *SONET/SDH Frame Generator User Guide*.

Testing 10 GbE devices

Ten Gigabit/s Ethernet transceivers are specified in an addendum to the IEEE 802.3 Ethernet specification. ParBERT provides a tool for testing the 10GBASE-R side as well as the XAUI side. The related document is the manual *Testing 10 Gbit/s Ethernet Devices*.

Recirculating optical loop tests

Recirculating loop tests are a method for simulating long-distance optical links in a laboratory. ParBERT systems can be used for performing such tests. The manual *Recirculating Optical Loop Tests* explains how optical loop tests can be set up.

What’s New?

This chapter gives an overview of the most important changes and enhancements of the Agilent 81250 Parallel Bit Error Ratio Tester.

The information is organized as follows:

Major Changes in Rev. 5.1

The focus of this revision is on:

- New optical data analyzer module
- New tool for generating VSR4-formatted data segments
- Application manual for setting up recirculating optical loop tests
- Special support for programming ParBERT systems via the LAN

New Data Analyzer Module

The new module can receive and analyze optical as well as electrical signals at data rates between 21 Mbit/s and 3.35 Gbit/s.

E4811A 3.35 Gbit / s data analyzer module

The E4811A data analyzer module has one channel. It includes an optical-to-electrical (O/E) converter and an electrical data analyzer. It is calibrated for 850 nm wavelength.

The O/E converter includes a sensor and a comparator. External electrical components like filters can be inserted between the sensor and the comparator. The O/E converter provides a differential signal to the data analyzer.

Benefits

This module makes it possible to test optical data transmitters without external O/E converters. In combination with the available E4810A data generator module, you can now test both sides of optical transceivers directly with ParBERT.

Such measurements are fully supported by the ParBERT Measurement Software.

For details see *"Modules" on page 43*.

Updated ParBERT Product Structure

The following table summarizes the presently supported hardware components and system configurations. The table is sorted according to frequency requirements.

You can identify the available modules and fitting frontends. Note that for data rates above 2.7 Gbit/s the high performance clock module E4808A is required.

Table 1 Supported Modules and Frontends

Max. data rate		Generator	Analyzer	Clock module	Comment
45 Gbit / s	**Modules:**	E4868B MUX	E4869B DEMUX	E4808A	ParBERT 43 / 45G special—comes with 8 data generator / analyzer modules
10.8 Gbit / s	**Modules:**	E4866A N4868A	E4867A	E4808A	No frontends—one channel per module Optional 10.8 GHz booster module; one channel standard, 2nd channel optional
3.35 Gbit / s optical	**Modules:**	E4810A	E4811A	E4808A	One channel per module Both modules can also be used for generating / analyzing electrical signals
3.35 Gbit / s	**Modules:**	E4861B	E4861B	E4808A	Two frontends (channels) per module
	Frontends:	E4862B	E4863B		
2.7 Gbit / s	**Modules:**	E4861A	E4861A	E4808A, E4805B	Two frontends per module
	Frontends:	E4862A	E4863A		
1.65 Gbit / s[a]	**Modules:**	E4861A	E4861A	E4808A, E4805B	Two frontends per module
	Frontends:	E4864A	E4865A		
675 Mbit / s	**Modules:**	E4832A	E4832A	E4808A, E4805B	Four frontends per module
	Frontends:	E4838A, E4843A	E4835A		E4835A means 2 frontends

[a] E4864A and E4865A frontends delivered before February, 2002, are confined to 1.35 Gbit / s.

For details please refer to the *Agilent 81250 ParBERT Technical Specifications*.

VSR4 Frame Generator

The VSR4 Frame Generator is a tool that allows you to generate ParBERT data segments that hold test data formatted according to the implementation agreement OIF-VSR4-01.0.

The payload can be pure PRWS (pseudo random data) or data formatted as SONET frames (generated with the SONET/SDH Frame Generator).

This supports the testing of components that conform to the VSR OC-192/STM-64 interface specifications.

The tool comes with online help and the manual *VSR4 Frame Generator*.

Support of Recirculating Optical Loop Tests

Recirculating loop tests are a method for simulating long-distance optical links in a laboratory. ParBERT systems can be used for performing such tests.

New Manual “Recirculating Optical Loop Tests”

The manual *Recirculating Optical Loop Tests* explains how optical loop tests can be set up with ParBERT.

Support for Programming ParBERT Systems via the LAN

The ParBERT software package includes libraries that provide multiple ways for users as well as for test and measurement applications to control and program ParBERT via the LAN.

New Manual “Agilent 81250 ParBERT LAN Programming Guide”

The *LAN Programming Guide* provides a practical guide to programming ParBERT via the LAN using Agilent IO Libraries, Telnet, Socket connections, or all of them. The theory can be found in the manuals enclosed in the Agilent IO Libraries install package.

Major Changes in Rev. 5.0

The highlights of this revision are:

- Optical data generator module
- New tool for testing 10 Gigabit/s Ethernet devices
- Enhancements of the ParBERT user interface
- Enhancements of the ParBERT Measurement Software
- A new method for aligning analyzer bit recognition to the incoming data
- Remote check of connector status

New Data Generator Module

The new module can generate optical as well as electrical signals at data rates between 21 Mbit/s and 3.35 Gbit/s. With 850 nm wavelength (option 001), the optical interface is in the short distance range.

E4810A 3.35 Gbit / s data generator module

The E4810A data generator module has one channel. It includes an electrical signal generator and an electrical-to-optical (E/O) converter. It features variable crossing and variable jitter. Jitter distribution can be controlled by an external voltage source.

Benefits

This module can be used for stimulating optical receivers. In combination with the available electrical analyzers (like the E4863B frontends), you can stimulate optical receivers and measure the electrical response.

Such measurements are fully supported by the ParBERT Measurement Software.

For details see *"Modules" on page 43*.

10Gb Ethernet Tool

The 10Gb Ethernet Tool supports tests of 10 Gbit/s Ethernet (10GbE) transceiver devices that conform to the 10GBASE-R and XAUI specifications as defined in the IEEE 802.3 addendum for 10GbE (IEEE 802.3ae).

Testing Xenpak devices

Such devices are standardized and developed, for example, by the members of the Xenpak authority.

The 10Gb Ethernet Tool allows you to generate ParBERT data segments holding 10GbE-formatted data as well as recommended test patterns. It enables you to execute 10GbE tests. It can automatically upload and post-process captured data.

New Manual "Testing 10 Gbit / s Ethernet Devices"

The information on 10GbE tests has been combined in the new application manual *Testing 10 Gbit/s Ethernet Devices*.

Enhancements of the ParBERT User Interface

The most important enhancement refers to the frequency setup.

The *segment resolution* is the length of a word in the memory of one or several ParBERT modules. Since the memory is organized to hold approximately 128 Kwords, the word length defines the available amount of memory.

Depending on the type of module and the chosen data rate, the segment resolution required for a particular port or channel can vary between one and 256 bits.

Setting the optimum segment resolution requires some insight into the way ParBERT generates bit rates (described in *"Frequency Multiplier and Segment Resolution" on page 68*).

Automatic calculation

To assist users who do not wish to dig into those details, the Parameter Editor for the clock module now allows you to choose between manual and automatic mode.

In automatic mode, the program automatically calculates a suitable segment resolution.

For details see *"How to Set the General System Frequency" on page 178*.

Enhancements of the ParBERT Measurement Software

The ParBERT Measurement Software includes the DUT Output Timing/Jitter Measurement.

Up till now, the jitter evaluation has only reported the total jitter (TJ).

DJ / RJ separation

New setup parameters and calculations have been added measurement to separate between random jitter (RJ) and deterministic jitter (DJ).

For details, please refer to the documentation of the Agilent 81250 ParBERT Measurement Software.

Fast Bit Synchronization

If pure PRxS data is used, Fast Bit Synchronization is a new way to synchronize analyzers to the incoming data stream.

This method is fast, because it needs only few data bits and does not optimize the sampling delay.

Fast Bit Synchronization does not replace the current synchronization methods. It does not report the final delay.

Fast Bit Synchronization has been implemented to support first of all "Recirculating Loop Tests" for optical fiber connections.

For details see *"Fast Bit Synchronization" on page 97*.

Remote Check of Connector Status

The frontends have built-in protection circuits which automatically disconnect a frontend if an attempt is made to operate the frontend under intolerable conditions.

If this happens, the user interface is neither informed nor updated. In case of a problem, it is therefore recommended to inspect the green LEDs above the frontend connectors. They indicate the physical connection status.

A new command has been implemented that allows remote programs to check the connector status.

For details see the *System Programming Guide and SCPI Reference*.

Major Changes in Rev. 4.5

This revision of the Agilent 81250 software provides:

- Extended support for testing SFI-5 SERDES devices
- Support of the enhanced ParBERT 43G systems
- Support for the new N4868A 10.8 GHz booster module
- Enhancements of the ParBERT Measurement Software

Enhanced Support for Testing SFI-5 Devices

The tools for testing SFI-5 SERDES devices (SERializers/DESerializers) are the SFI-5 Frame Generator utility and the SFI-5 Post Processing Tool.

The SFI-5 Frame Generator utility is used for generating SFI-5 formatted data segments, and the SFI-5 Post Processing Tool for analyzing the captured data.

SFI5 Data Segment

For E4861B 3.35 Gbit/s data generator/analyzer modules, a new type of data segment is provided. It specifies pure PRWS data that is formatted according to the SFI-5 standard and includes the DSC signal.

The advantage is that such a segment does not have to be generated on the ParBERT controller and downloaded—the E4861B modules generate the data by themselves, the random data as well as the deskew channel data. This saves not only test time and data memory, but allows the use of high-order PRxS polynomials up to 2^{31}–1.

New Manual “Testing SFI-5 Devices”

To reflect Agilent’s ongoing effort to provide not only general purpose instruments but also turnkey solutions, the information on SFI-5 has been combined in the new application manual *Testing SFI-5 Devices*.

Enhanced ParBERT 43G Systems

The MUX/DEMUX modules of the high-speed ParBERT 43G systems have been enhanced. New ParBERT 43G bundles include these modules.

New bundles

You can now choose between four ParBERT 43/45G bundles. They contain either 2.7 Gbit/s or 3.35 Gbit/s data modules and frontends.

For details see *“ParBERT 43/45G Components” on page 108*.

New E4868B MUX / E4869B DEMUX modules

The B-versions of the MUX/DEMUX modules provide additional and extended features that can be controlled from the user interface.

The E4868B MUX module allows you to add an offset voltage to its output. The E4869B DEMUX module allows you to tune its sampling delay manually.

Both modules support full segment resolution and hence memory capacity of the 3.35 Gbit/s data generators/analyzers. A system equipped with these modules can exceed frequencies of 43.2 Gbit/s.

For details see *“How to Set Up a 43G MUX/DEMUX Module” on page 210*.

New Module

New N4868A 10.8 GHz booster module

The N4868A 10.8 GHz booster module is an add-on to the E4866A 10.8 Gbit/s data generator modules. It contains two or four signal conditioners/amplifiers for applications that require optimum signal quality.

The N4868A 10.8 GHz booster module improves the slew rates and hence reduces the transition times of the generated signals considerably.

This module can be connected between the E4866A 10.8 Gbit/s data generator module and the DUT. Its input is the differential output of one or two generator modules.

You can also use the high-speed amplifiers in single-ended mode. In this mode, you can operate each amplifier separately. For example, you can connect two 10.8 Gbit/s generators—both set to single-ended operation—and amplify two separate signals.

The software supports individual parameter settings for each of the amplifiers. For details see *"N4868A Module" on page 51*.

New Output Level Measurement

A new measurement type has been added to the ParBERT Measurement Software. The Output Level measurement allows you to determine the bit error rate at varying receiver thresholds.

This measurement can be used for investigating the behavior of the DUT. A direct result is the determination of the optimum analyzer threshold level for receiving data from the DUT with maximum confidence.

Variable Decision Threshold Method

The method used by this measurement is commonly known as *Variable Decision Threshold Method*. It provides a "vertical" analysis of the eye opening seen by the receiver(s). This method allows you to determine more than just the actual levels.

The Output Level measurement calculates also the Q-factor (a measure that describes the quality of the received signal) and derived values. These results can assist you in characterizing the device. They can also enable you to predict very low bit error rates that would take a long time to measure.

For details please refer to the documentation of the Agilent 81250 ParBERT Measurement Software.

Major Changes in Rev. 4.3

This revision of the Agilent 81250 user software provides first of all enhancements and extended support of ParBERT 43G systems.

ParBERT 43 / 45G Systems with 3.35 GHz Modules

Up to now, the ParBERT 43G systems were restricted to E4861A 2.7 Gbit/s data generator/analyzer modules, equipped with E4862A generator or E4863A analyzer frontends.

From now on, ParBERT 43/45G systems can also be based on E4861B 3.35 Gbit/s data generator/analyzer modules.

A ParBERT 45G pattern generator system would use eight E4861B data generator/analyzer modules with 16 E4862B generator frontends connected to the MUX module.

A ParBERT 45G error detector system would use eight E4861B data generator/analyzer modules with 16 E4863B analyzer frontends connected to the DEMUX module.

Software Enhancements

New size of the main window

When you start this revision of the ParBERT user software for the first time, you will find that the main window has been enlarged. This supports the display of the ParBERT 43G windows, especially the Connection Editor.

This requires a minimum display resolution of 800 × 600 pixels.

New predefined level

A new predefined signal level has been added. It is called CML and is meant (and automatically set) for channels connected to E4868 MUX or E4869 DEMUX modules.

DEMUX rewiring

Depending on the complexity of the demultiplexer and the chosen rewiring algorithm, DEMUX rewiring can take some time. It now runs with twice the speed as before. In addition, the *Rewiring* phase is now indicated by the side of the Run/Stop buttons, like all the other phases of a test.

Major Changes in Rev. 4.2

Revision 4.2 of the Agilent 81250 user software supports the ParBERT module and frontends for data generation and analysis at rates from 21 Mbit/s up to 3.35 Gbit/s.

In addition, the frontends E4864A and E4865A have been enhanced.

New Data Generator / Analyzer Module

The new module is:

- E4861B 3.35 Gbit/s data generator/analyzer module

Like the E4861A 2.7 Gbit/s data generator/analyzer module, this new module houses two frontends. Generator and analyzer frontends can be mixed.

New Data Generator / Analyzer Frontends

The new frontends that can be installed into the E4861B data generator/analyzer module are:

- E4862B 3.35 Gbit/s data generator frontend
- E4863B 3.35 Gbit/s data analyzer frontend

Both frontends have a maximum data memory capacity of 16 Mbit.

E4862B data generator

In addition to the extended frequency range and memory, the E4862B 3.35 Gbit/s data generator frontend has features that are not available for the other generators:

- Variable signal crossing

  Standard data generators produce a signal where the crossing point is at 50 % of the amplitude. For the E4862B data generator frontend, the crossing point of NRZ signals can be freely adjusted between 30 % and 70 % of the signal amplitude. This allows you to generate signals with non-symmetrical eye openings.

  For details see *"How to Set Generator Timing Parameters" on page 227*.

- Voltage controlled delay

  The E4862B data generator frontend has an input connector. This connector allows you to connect a voltage that controls the signal delay. The linear voltage range is ±0.5 V. By varying this voltage, it is possible to generate a signal with jitter of arbitrary distribution.

Frequencies from DC to 200 MHz are supported for the delay control.

For details see *"How to Use the Variable Delay (FEs of Type E4862B Only)" on page 238*.

- Precision clock generation

  When the E4862B data generator is connected to a pulse port, it produces a clock signal, like all the generators. For this frontend, you can now choose between *Pulse Mode* and *Clock Mode*.

  In *Pulse Mode*, you can specify a start delay, the data format, and also width or duty cycle. In *Clock Mode*, a precision clock signal is generated, with fixed delay, 50 % duty cycle, and very low jitter.

  For details see *"How to Set Generator Timing Parameters" on page 227*.

- RZ and R1 data formats

  The E4862B data generator supports not only the NRZ data format, like the E4862A data generator frontend does, but also the RZ and R1 data formats.

E4863B data analyzer

The E4863B 3.35 Gbit/s data analyzer frontend has a higher speed and more memory than the E4863A 2.7 Gbit/s data analyzer.

Apart from this, it has the same capabilities:

- It captures and analyzes incoming data in real time
- It provides an AUX OUT connector that delivers a recovered clock
- It supports all the measurements provided by the Agilent 81250 Measurement Software

Benefits

With the new module and frontends, the Agilent 81250 Parallel Bit Error Ratio Tester addresses:

- 10 Gigabit Ethernet (10GbE) components, including 10G Ethernet Attachment Unit Interfaces (XAUI, 4 × 3.125 Gbit/s) as proposed by the Xenpak Multisource Agreement
- MUX / DEMUX components (OC-48)
- Multiple electrical-to-optical (E/O) and optical-to-electrical (O/E) transceivers (VSR OC-768)
- Cross-point switching components

Enhanced Data Generator / Analyzer Frontends

An E4861A data module can accommodate the frontends E4864A and E4865A. New frontends of this type now cover a wider frequency range:

- E4864A data generator: Formerly up to 1.35 Gbit/s—now up to 1.65 Gbit/s
- E4865A data analyzer: Formerly up to 1.35 Gbit/s—now up to 1.65 Gbit/s

If new E4864A or E4865A frontends have been added to your system, it is recommended to check the frontend labels that indicate the maximum frequency. The older frontends remain limited to data rates of 1.35 Gbit/s.

Updated ParBERT Product Structure

The following table summarizes the presently supported hardware components. The table is sorted according to frequency requirements.

Table 2 Supported Modules and Frontends

Max. data rate		Generator	Analyzer	Clock module	Comment
43.2 Gbit / s	**Modules:**	E4868A MUX	E4869A DEMUX	E4808A	ParBERT 43G special
10.8 Gbit / s	**Modules:**	E4866A	E4867A	E4808A	No frontends—one module per channel
3.35 Gbit / s	**Modules:**	E4861B	E4861B	E4808A	Two frontends per module
	Frontends:	E4862B	E4863B		
2.7 Gbit / s	**Modules:**	E4861A	E4861A	E4808A, E4805B	Two frontends per module
	Frontends:	E4862A	E4863A		
1.65 Gbit / s[a]	**Modules:**	E4861A	E4861A	E4808A, E4805B	Two frontends per module
	Frontends:	E4864A	E4865A		
675 Mbit / s	**Modules:**	E4832A	E4832A	E4808A, E4805B	Four frontends per module
	Frontends:	E4838A, E4843A	E4835A		E4835A means 2 frontends

[a] E4864A and E4865A frontends delivered before February, 2002, are confined to 1.35 Gbit / s.

For details please refer to the *Agilent 81250 ParBERT Technical Specifications*.

Software Enhancements

Analyzing a received SFI-5 pattern is not trivial. Before the bit error rate can be measured, the skew between the 16 data lines and the 17th data channel must be considered to ensure that the received PRBS data is valid.

SFI-5 Post Processing Tool

For this purpose, the SFI-5 Post Processing Tool has been developed.

This tool analyzes captured SFI-5 patterns. It determines the skew and calculates the BER not only of the 16 data lines but also of the additional synchronization channel.

The SFI-5 Post Processing Tool comes with its own user interface. This interface allows you to start a test, to inspect the results, and to save the captured data into a data segment. For details see the manual *Testing SFI-5 Devices*.

Major Changes in Rev. 4.1

Revision 4.1 of the Agilent 81250 user software supports the ParBERT 10.8 Gbit/s data generator/analyzer modules. It provides also enhancements of the Segment Editor and a new way to switch the connectors on and off.

New Data Generator / Analyzer Modules

The new modules are:

- E4866A 10.8 Gbit/s data generator module
- E4867A 10.8 Gbit/s data analyzer module

These modules are capable of creating or analyzing data streams at 9.5 Gbit/s up to 10.8 Gbit/s.

Benefits

You may think of testing a multiplexer component, stimulated by four 2.7 Gbit/s generators or eight 1.35 Gbit/s generators. One single E4867A 10.8 Gbit/s analyzer can measure the serial response. Similarly, a demultiplexer can be stimulated by an E4866A 10.8 Gbit/s data generator and analyzed by four, eight, or 16 lower speed analyzers.

You may also think of high-speed applications. For example, four E4866A data generator modules can be used for stimulating an OC-768 4:1 multiplexer. The serial output of this multiplexer with a data rate around 40 Gbit/s can be analyzed by a ParBERT 43G error detector system.

Using a ParBERT 43G pattern generator system, you can send a serial bit stream of up to 43.2 Gbit/s to an OC-768 demultiplexer component. Using four E4867A data analyzer modules, it is now possible to analyze the parallel output of a 1:4 demultiplexer.

Details

These high-speed modules do not have frontends. Each has one pair of differential connectors which is identified by the software as one connector. For these modules, one channel means one module.

From the user's point of view, this is the only remarkable difference to the lower speed modules. For the connection and parameter setup, the same functions of the Connection Editor and Parameter Editor are used.

Automatic analyzer sampling delay adjustment and all the measurements provided by the Agilent 81250 Measurement Software can be used with the E4867A 10.8 Gbit/s data analyzer modules.

Enhancements of the Segment Editor

For testing a multiplexer, one needs parallel generated and serial expected data. Testing a demultiplexer requires serial generated and parallel expected data. The data contents is the same on both sides—only the data distribution changes.

To support the setup of data to be generated and expected for multiplexer and demultiplexer tests, three functions have been added to the Segment Editor, and one has been enhanced.

Deserialize

The Deserialize function allows you to convert a serial bit stream to parallel format. This is done in the data segment. A segment holding serial data contains just one trace. With the Deserialize function, this data can be split into an arbitrary number of traces.

This, for example, makes it easy to create the data expected from a demultiplexer. To adapt to the characteristics of the demultiplexer, the function provides several alternatives for sorting the data.

Serialize

This function is the counterpart of the deserialize function. It is used to convert parallel data to serial. A segment holding multiple traces is converted to a serial segment holding one trace.

Again, several alternatives are provided for sorting the data.

Find

Formatted data is generally organized in blocks. Such blocks contain header, control, and payload data. You may wish to pursue the order of these blocks.

This is supported by the Find function. This function allows you to search for a certain bit combination within a segment. You can search for parallel and serial bit patterns.

Enhanced copy and paste

When testing n:1 multiplexers or 1:n demultiplexers, you may need separate ParBERT systems on the generating and analyzing sides. This depends on the frequency ratio. Using two ParBERT systems, in turn, means using two user interfaces.

You may, for example, have set up suitable test data for the generator system. But how to transfer that data to the analyzer system, so that it recognizes the expected data?

This is covered by the enhanced copy and paste functions of the Segment Editor.

As soon as the Segment Editors of both user interfaces are open, you can copy and paste memory data from one to the other.

This, combined with the Deserialize/Serialize functions, makes it easy to set up all kinds of multiplexer/demultiplexer tests.

Switching Connectors ON / OFF

The tool bar of the ParBERT user interface has always had a Connectors Off/On button for disconnecting all the frontends and for re-establishing the previous connections. This was done by switching relays in the frontends.

For frontends with data rates above 675 Gbit/s, this behavior can be changed. It is now possible to specify whether the relays shall be switched or whether the frontends shall be disconnected by grounding.

Especially in a production environment, grounding is a way to increase the lifetime of the ParBERT relays.

Major Changes in Rev. 4.0

Revision 4 supports not only all common ParBERT configurations but also the *Agilent 81250 ParBERT 43G Systems*. This is accompanied by enhancements of the Agilent 81250 Configuration Tool and the start procedure of the Agilent 81250 user software.

Agilent ParBERT 43G Systems

The Agilent 81250 ParBERT 43G is a solution for generating and analyzing electrical data streams of 38 Gbit/s up to 43.2 Gbit/s.

It allows you to stimulate and analyze 16:1 multiplexers and 1:16 demultiplexers at data rates of 2.7 Gbit/s and 43.2 Gbit/s, according to the OC-768 and SFI-5 (Serdes Framer Interface 5) data range.

It allows you also to determine the bit error rate of serial devices or transmission lines operated at 43.2 Gbit/s. It supports the investigation of FEC devices at 43.01841 Gbit/s including the FEC rate resulting from 255/236 overhead.

The Agilent 81250 ParBERT 43G systems are offered as bundles—one for pattern generation and one for error detection. But it is also possible to upgrade existing systems.

A setup for testing both multiplexers and demultiplexers would use both bundles. The bundles include also the E4808A clock module which is superior to the well-known E4805B clock module.

For details see *“ParBERT 43/45G Systems” on page 107*.

Software Enhancements

An Agilent 81250 ParBERT 43G system is automatically recognized and the user software comes up with a Connection Editor which is preconfigured.

Using both Agilent 81250 ParBERT 43G bundles means using two ParBERT user interfaces. This is required for independent clock generation and parameter setup. The capability to run more than one user interfaces has always been there.

Support of multiple user interfaces

The Agilent 81250 Configuration Tool and the ParBERT user software now greatly support the use of several user interfaces:

- More than one user interface can be automatically started.
- Every user interface can be individually configured.

  The configuration parameters for each user interface include: Location of the firmware server (local or LAN address), name of the system to be operated, name of the setting to be automatically downloaded to the system.
- Every user interface can easily be switched to operate one of the configured systems.
- Tests can be started an stopped simultaneously on two or more systems.

  For details on the System Starters see *"How to Use the System Starter Utilities" on page 385*.

Data generation for SONET and SFI-5

The Utilities panel provides also the access to two tools used for generating data segments for the ParBERT 43G.

- The SONET SDH Frame Generator generates data that is formatted according to SONET standards (STS 3 to STS 768) or SDH (STM 1 to STM 256). This tool includes its own online help.
- The SFI-5 Frame Generator generates data that is formatted according to SFI-5 standards. For details, see *SFI-5 Frame Generator and the SFI-5 Data Segment* in the manual *Testing SFI-5 Devices*.

External clock sources

If an external clock source is connected to the master clock module, then this source determines the system clock frequency. The software now provides not only the clock multiplier but also a clock divider.

This means, you can, for example, set the system clock frequency to 3/5 of the external clock frequency, or to any other arbitrary value.

PLL lock indicator

When an external clock source is used, the phase locked loop of the master clock module locks onto that source. The PLL lock indicator informs you in case the clock system could not lock or has lost its synchronization.

Automatic Rewiring of Demultiplexer Terminals

When testing demultiplexers, you apply serial data to one terminal and analyze parallel data from a number of terminals. Pure PRBS and PRWS data are generally well suited for testing demultiplexers.

But if you are using memory-based data, you may encounter the problem that the data from the DUT is correct, but does not start with the first terminal—the one numbered “1”. This, however, is required for comparing the received data with the stored data segment. Trace 1 of the segment is expected from the first terminal, trace 2 from the next, etc.

The ParBERT now provides a means to fully automatically check the incoming data and rearrange the sequence of the terminals. As a result, the first terminal is the one that delivers the first bit of every word, the second terminal delivers the second bit, and so on, and the incoming data can be compared with the expected.

For details see *“Automatic Rewiring of Demultiplexer Terminals” on page 119*.

Introduction to the System

This chapter familiarizes you with the Agilent 81250 Parallel Bit Error Ratio Tester—its components, operating principles, and terms.

The information is organized as follows:

- *"System Capabilities" on page 38*—a summary of the system's purpose and technical highlights
- *"System Components" on page 40*—the description of the ParBERT hardware components
- *"Operating Principles" on page 58*—the philosophy of virtual systems and setup models and its impact on the software structure
- *"Timing Principles" on page 66*—about clock sources and how the system clock is generated by frequency multiplication
- *"Data Generation Principles" on page 78*—the characteristics of data sequences, data blocks, and data segments
- *"Principles of Analyzer Sampling Point Adjustment" on page 87*—the three methods for adjusting the analyzer sampling delay
- *"Data Capturing and Analysis Principles" on page 101*—a description of the four standard functional tests and their result displays
- *"Event Handling Principles" on page 104*—the explanation of how the system can be influenced by internal and external events
- *"ParBERT 43/45G Systems" on page 107*—an introduction to the ParBERT 43/45G that allows to create and analyze data streams at rates of 43.2 Gbit/s and higher
- *"Automatic Rewiring of Demultiplexer Terminals" on page 119*—an overview and more of special features for testing demultiplexers

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If it has been installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial*.
If not, you can download it from the web through
http://www.agilent.com/find/81250demo

System Capabilities

The Agilent 81250 Parallel Bit Error Ratio Tester is first of all meant for testing high-speed data communication equipment (DCE), but can also be used as a multi-purpose digital stimulus/response (DSR) system.

The system can be operated from the graphical user interface or controlled via LAN or GPIB, for example in an automated test rack. It can also control other instruments via GPIB. It has programming interfaces to VEE, C/C++, and others.

Verify and Characterize Digital Devices

The device under test (DUT) and its test setup are mirrored in the software.

The graphical user interface shows a raw DUT template in the Connection Editor window. In this window it is possible to define groups of signals for the DUT—these groups of signals are called “ports”. The DUT template offers two types of ports:

- Data ports are used for data signals such as stimulus data and expected response data.
- Pulse ports are used for pure parametric signals such as clock signals.

All signal parameters can be set up conveniently for a group of pins (a port) as well as separately for single DUT pins (terminals).

The system has data generating and analyzing frontends.

Cable delays and signal skew in the test setup can be compensated by using the deskew feature.

Key Features

The following list summarizes the most important features:

- Stimulus as required, real-time error analysis and margin tests
- Parallel data rates of generator and analyzer channels up to 10.8 Gbit/s
- Serial data rates up to 45 Gbit/s (ParBERT 43/45G)
- Up to 32 Mbit data memory per channel
- Up to 128 generator/analyzer channels at 675 Mbit/s. Up to 64 channels at 1.35, 2.7, or 3.35 Gbit/s. Up to 32 channels at 10.8 Gbit/s
- 2 ps timing resolution
- Pattern formats NRZ, DNRZ, RZ, R1
- Pseudo random bit and word streams (PRBS/PRWS) up to $2^{15}-1$ plus $2^{23}-1$ and $2^{31}-1$
- Automatic analyzer sampling delay adjustment
- Sequencing with up to five loop levels (nested loops)
- Variable delay, width, transition times, voltage levels—individually adjustable for each frontend
- Logical XOR addition of two or four 675 MHz generator channels
- Analog voltage addition of two 675 MHz generator channels
- Semi-automatic signal delay compensation (deskew)
- Event recognition and reactions upon events
- Tabular and graphical result presentation
- Automated measurements based on measuring the bit error rate at many points in time and voltage: Eye Opening, Fast Eye Mask, DUT Output Timing/Jitter.
- Pass/fail measurements
- Modular hardware structure—mix of low and high-speed channels
- Frontends for differential and low voltage signals
- Comprehensive support for testing multiplexers/demultiplexers—multiple frequencies, internal and external clocks, DEMUX rewiring, data deserialize and serialize functions
- Comprehensive support for testing optical transceivers—optical signal generators/analyzers; tools for creating formatted generated and expected data

For details please refer to the *Agilent 81250 Technical Specifications*.

System Components

The Agilent 81250 Parallel Bit Error Ratio Tester is available in many configurations.

Standard configurations include:

- *"Mainframes and Controllers" on page 40*—the controller can be an external PC or workstation, or a 2-slot VXI module embedded into the mainframe
- *"Modules" on page 43*—a system consists of at least one clock module and one or several data generator/analyzer modules
- *"Frontends" on page 53*—frontends generate or receive signals. They are plugged into data generator/analyzer modules
- *"Trigger Pod" on page 55*—a special device that allows to recognize external events

NOTE This section describes the basic components. For information on the ParBERT 43/45G systems please refer to *"ParBERT 43/45G Systems" on page 107*.

Mainframes and Controllers

An Agilent 81250 Parallel Bit Error Ratio Tester consists of a VXI mainframe, a controller, and modules plugged into the mainframe.

Figure 2 E4860A System Configuration

Mainframes

13-slot Mainframe

The standard mainframe is the E4803A VXI mainframe with 13 VXI slots.

Figure 3 E4803A VXI Mainframe with Embedded Controller and Modules

Up to two expander frames can be added. The first E4848B expander frame consists of an E8403A mainframe, two E1482B VXI bus extender modules, and connection cables.

The VXI bus extender modules require one slot in both the basic and the added frame. A second expander frame needs only one E1482B VXI bus extender module.

If an external controller is used (see *“Controller Options” on page 41*), then the E8491B IEEE 1394 PC link to VXI replaces the VXI bus extender modules.

Controller Options

Two options are available for controlling the system:

External ParBERT Controller

- The E8491B IEEE 1394 PC link to VXI (opt. #013)

  This option allows to use an external PC running under Windows NT or Windows 2000 as the system controller. The option includes a PCI board to be installed in the computer, a 1-slot VXI module to be installed in the mainframe, and all required software.

  Opt. #013 leaves 12 mainframe slots for modules.

Embedded ParBERT Controller

- The built-in 2-slot VXI controller E9850A (opt. #012)

  This controller is a VXI module that includes harddisk, diskette drive, a serial and a parallel interface, SCSI controller, GPIB and LAN interface. Operating system and ParBERT software are already installed.

  Opt. #012 leaves 11 mainframe slots for modules.

  Opt. #012 requires a monitor, keyboard and mouse (supplied with Opt. #010).

Open VXI Configurations

The Agilent 81200 Data Generator/Analyzer Platform also supports Open VXI configurations. Open VXI enables you to set up your test equipment as compact as required.

As long as free slots are available, VXI modules of other systems can be plugged into the Agilent 81250 mainframe. You need only take care that the Agilent 81250 modules start from the leftmost slot and remain in contiguous slots. Additional software can be installed on the built-in harddisk or the external PC to operate these modules.

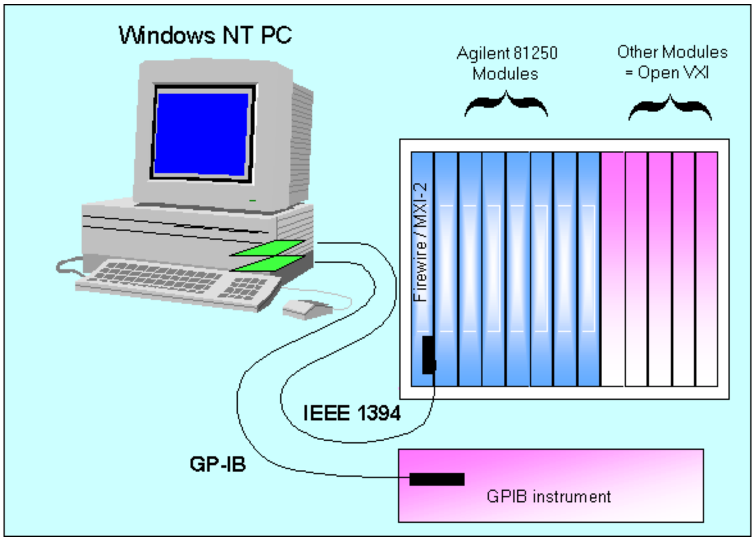


Figure 4 IEEE 1394 PC Link and Open VXI Configuration

For details please refer to the *Agilent 81250 Installation Guide* (available as PDF) and the *Configuration Guide*.

Modules

An Agilent 81250 Parallel Bit Error Ratio Tester is comprised of at least one clock module and one data generator/analyzer module.

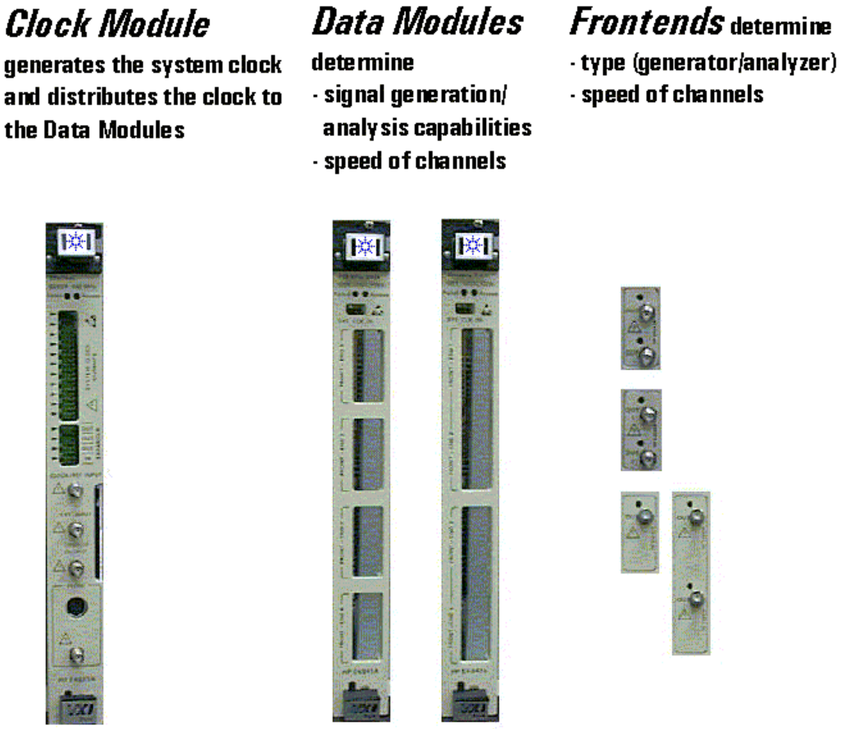

Clock Module
generates the system clock and distributes the clock to the Data Modules
Data Modules
determine
- signal generation/ analysis capabilities
- speed of channels
Frontends determine
- type (generator/analyzer)
- speed of channels

Figure 5 Modules and Frontends

Clock Modules

The master clock module generates the system clock and synchronizes all data generator and analyzer channels of a system.

The master clock module provides the sequencing capability of a system and can use either its internal synthesized clock source or an external clock source. The internal clock synthesis can be locked to a common frequency standard using the PLL reference input.

The following clock modules are supported:

E4805B Clock Module

- E4805B Clock Module

  This module synchronizes up to 11 data analyzer/generator modules. It can additionally control up to two slave clock modules. An expander frame requires a slave clock module.

A deskew probe can be connected and the Agilent 81200 Trigger Pod can be attached to the master clock module.

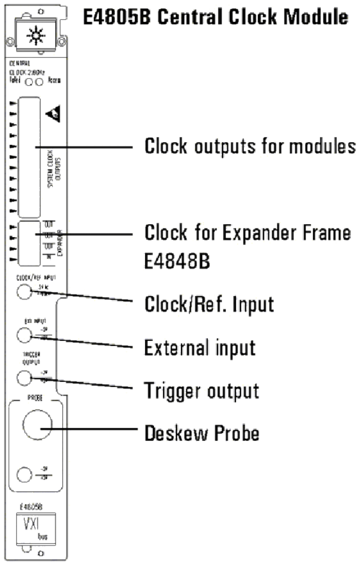

E4805B Central Clock Module
Clock outputs for modules
Clock for Expander Frame E4848B
Clock/Ref. Input
External input
Trigger output
Deskew Probe
VXI bus

Figure 6 **E4805B Clock Module**

The E4805B clock module permits the use of up to 5 loop levels within generated data sequences.

It is also possible to install two or more master clock modules in one mainframe. This results in mutually independent ParBERT systems that share only the housing (see also *"Virtual Systems" on page 58*).

E4808A Clock Module

- E4808A High Performance Clock Module

  This clock module has the same properties as the E4805B clock module, but superior jitter and noise characteristics.

  The E4808A clock module has to be used in all ParBERT systems that generate/analyze signals at rates above 2.7 GHz.

Data Generator / Analyzer Modules

Some data generator/analyzer modules have slots for two or four replaceable frontends.

There are frontends for generating and sourcing signals to the DUT and others for capturing and analyzing signals from the DUT.

Any combination of generator and analyzer frontends within a module is possible. However, it is generally recommended to install the generator and analyzer frontends in separate modules. This supports the concept of grouping output and input signals into "ports" and generating or expecting pseudo random word stream signals (PRWS).

NOTE There are also specific generator modules and analyzer modules. Such modules have no frontends that can be replaced.

The following data generator/analyzer modules are used:

E4832A Module

- E4832A Data Generator/Analyzer Module (up to 675 MHz):

  This module provides four slots for four generator and/or analyzer frontends.

  The E4832A module has a memory capacity of up to 2 Mbit per channel. It supports tests and measurements at data rates up to 675 Gbit/s.

E4861A Module

- E4861A Data Generator/Analyzer Module (up to 2.7 GHz):

  This module provides two slots for two frontends with maximum data rates of up to 1.65 Gbit/s or 2.7 Gbit/s. It provides a memory capacity of up to 8 Mbit per channel.

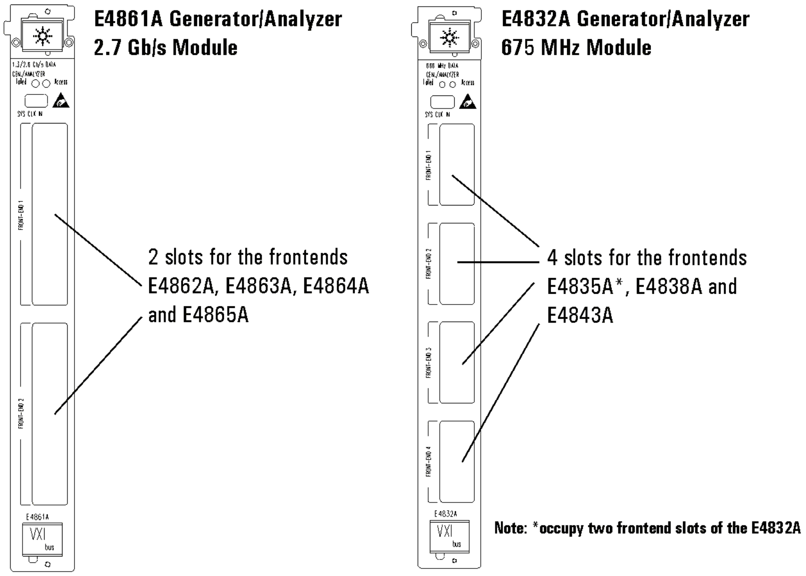


Figure 7 E4832A / E4861A Data Generator / Analyzer Modules

NOTE Your system may include older E4832A/E4861A data generator/analyzer modules, marked for frequencies of 1.33/2.67 GHz or 667 MHz. If this is the case, the achievable data rates are limited by the older modules.

E4861B Module

- E4861B Data Generator/Analyzer Module (up to 3.35 GHz):

  This module looks like the E4861A module. It provides two slots for two frontends with maximum data rates of up to 3.35 Gbit/s. Compared with the E4861A module, it has a much wider frequency range and twice the memory capacity.

  E4861B modules must only be used in conjunction with an E4808A clock module.

E4810A / E4811A Modules

- These modules are capable of generating or analyzing electrical or optical data streams at 21 Mbit/s up to 3.35 Gbit/s. They are:

  – E4810A 3.35 GHz electrical-optical data generator module

  – E4811A 3.35 GHz optical-electrical data analyzer module

  They can only be used in conjunction with an E4808A clock module.

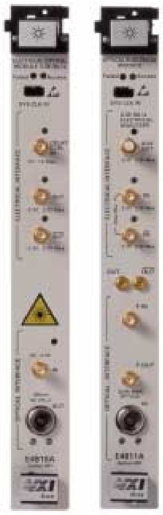


Figure 8 E4810A / E4811A Data Generator / Analyzer Modules

These modules have no replaceable frontends. For these modules, one *channel* means one module.

You can switch between optical and electrical operation (see *“How to Set the Mode of an Optical-Electrical Connector” on page 209*).

In electrical mode, connectors can be aligned and cables can be deskewed as usual (see *“How to Compensate for Internal and External Delays” on page 364*). The electrical connectors are protected as usual against overvoltage or short-circuit (see also *"Frontends" on page 53*). Watch the green LEDs.

In optical mode, you need an optical oscilloscope to measure the signal delays at the DUT (see *"How to Deskew Optical Connections" on page 374*).

Special E4810A characteristics

This generator module consists of an electrical signal generator and an electrical-to-optical (E/O) converter.

A short cable is used to connect the OUT connector of the signal generator to the E/O converter. Once it is enabled by software, the E/O converter produces the optical signal.

DANGER

The laser source is classified as Class 1M according to IEC 60825-1 (2001). It generates invisible infrared radiation. Do not look directly into the optical connector or the open end of a connected fiber when the laser is active.

NOTE

The laser is disabled if its mean power exceeds the specified capacity. This can happen, if unbalanced data is sent (for example long runs of pure zeros or pure ones). Watch the green LED of the E/O converter—it indicates the present status.

TIP

Remote programs can query the connector status.

For information on how to specify the optical level parameters, see *"How to Set Optical Generator Levels" on page 236*.

The signal generator can also be used without the E/O converter. It has the same advanced features as the E4862B, 3.35 GHz, differential output frontend with variable delay and variable signal crossing.

Special E4811A characteristics

The analyzer module consists of an optical-to-electrical (O/E) converter and an electrical data analyzer.

The O/E converter has an external cable connection between the optical sensor and the comparator. This makes it possible to insert an external electrical filter between the two components.

The O/E converter has also an ADC that allows to measure the average optical power and to set the decision threshold automatically.

Two short cables are used to connect the comparator outputs of the O/E converter to the differential inputs of the signal analyzer. The connections are illustrated in the figure below.

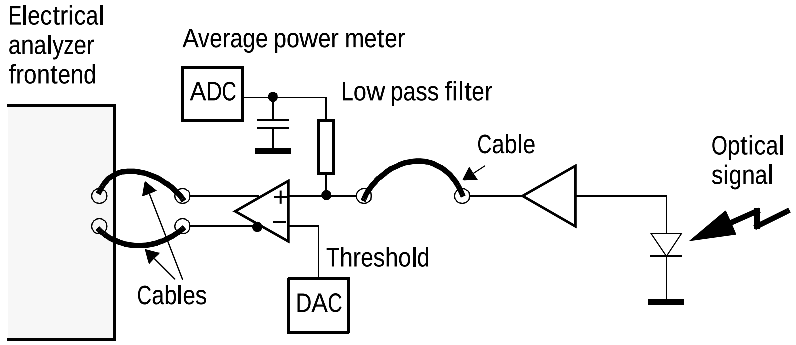


Figure 9 **E4811A O/E Converter Block Diagram**

In optical mode, the electrical analyzer receives a binary signal. The normal output of the O/E converter is connected to the normal input of the analyzer, the complementary output to the inverted input.

For information on how to specify the optical level parameters, see *"How to Set Optical Analyzer Levels" on page 248.*

The electrical signal analyzer can also be used without the O/E converter. It has the same advanced features as the E4863B, 3.35 Gsa/s, differential/single-ended analyzer frontend, including the auxiliary output that can be used for conditioning a recovered clock.

E4866A / E4867A Modules

- These modules are capable of creating or analyzing data streams at 9.5 Gbit/s up to 10.8 Gbit/s. They are:
  - E4866A 10.8 Gbit/s data generator module
  - E4867A 10.8 Gbit/s data analyzer module

  They can only be used in conjunction with an E4808A clock module.

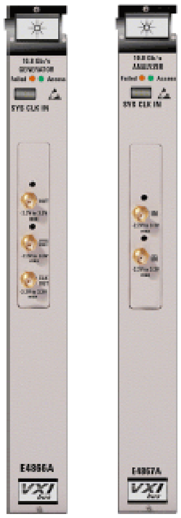


Figure 10 E4866A / E4867A 10.8 Gbit / s Data Generator / Analyzer Modules

These high-speed modules do not have replaceable frontends. Each has one pair of differential connectors which is identified by the software as one connector. For these modules, one *channel* means one module.

Both modules require a highly stable clock signal. If you are using an external source clock, make sure the clock source is stable enough to achieve a good jitter performance.

Special E4866A characteristics

The TRIGGER OUTPUT of an E4808A clock module is not stable enough to provide a suitable clock signal to a separate 10.8 G analyzer system.

For this purpose, the E4866A module has a CLK OUT connector. This connector provides a single-ended clock pulse of 9.5 GHz to 10.8 GHz which can be connected to the DUT or a separate analyzer system (using a 50 Ω impedance—for details see the *Agilent 81250 Technical Specifications*).

Note also, that unused CLK OUT connectors have to be terminated. A 50 Ω termination plug is delivered with the module. Signal performance degrades, if the CLK OUT connector is not properly terminated.

Special E4867A characteristics

The 10.8 Gbit/s analyzer E4867A has the block diagram shown below:

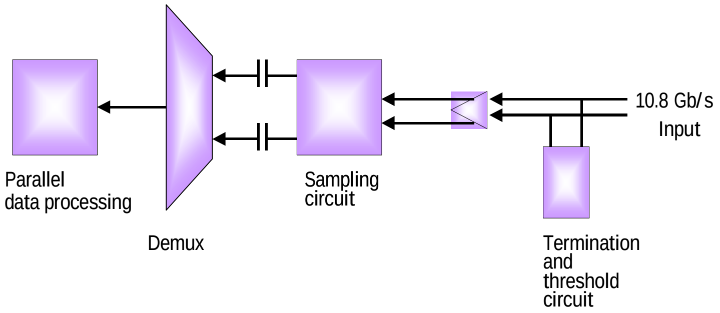


Figure 11 10.8 Gbit / s Data Analyzer Module Block Diagram

The E4867A has an internal AC coupling behind the sampling circuit. This is no problem for usual data communication, which is generally simulated using PRBS/PRWS.

However, when testing a device with the E4867A, long periods of zeros or ones have to be avoided.

The E4867A handles well:

- Pure PRBS
- Data that conforms to the following rule: On the average, the ratio of ones or zeros to the total number of bits must be within 9/19 (47.4 %) and 11/19 (57.8 %)
- Memory data with bursts of pure zeros or ones below 20,000 bits or 2 µs. On the average, the rule given above has to be observed.

After more than 2 µs of non-activity, the module has to recover. This takes up to 200 µs. This happens each time a test is started.

TIP Consider the following when setting up the test sequence:

- Enable automatic analyzer delay adjustment (see *"Automatic Delay Alignment" on page 90* and *"Automatic Bit Synchronization" on page 92*) and use suitable test data as described above. In this case, the settling time after starting the test is automatically considered.
- If automatic analyzer delay adjustment is not desired or possible, use an initialization block at the beginning of the sequence. This block must contain "balanced" data as described above and cover the settling time. At a data rate of 10 Gbit/s, at least 2 million bits should be generated and received.

- Ensure also that the data used for the subsequent test does not violate the limits stated above.

N4868A Module

The N4868A 10.8 Gbit/s Booster Module is an add-on to the E4866A 10.8 Gbit/s data generator module.

It is used to reduce the transition times of the generated signals and can be connected between the E4866A data generator and the DUT.

The E4866A 10.8 Gbit/s data generator module may produce an eye diagram like the one shown below:

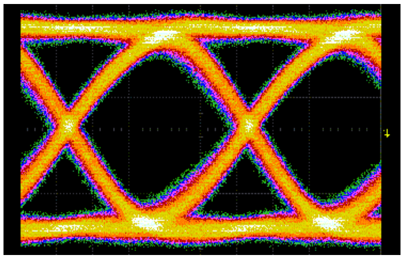


Figure 12 **10.8 Gbit / s Data Generator Output Example**

This is adequate for many applications. The booster module can convert this to the following:

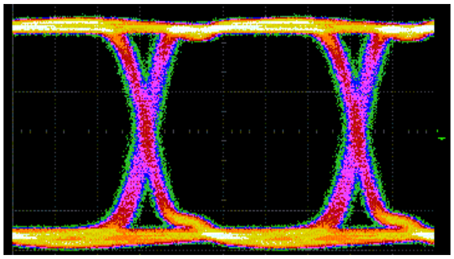


Figure 13 **10.8 Gbit / s Booster Output Example**

This yields a signal that is less prone to transmission errors.

The booster module can be plugged into any free adjacent slot of the system.

Remember that a ParBERT system, controlled by a master clock module (that is clockgroup one), does not tolerate empty slots (except between clockgroups, controlled by slave clock modules).

Using two matched cables, the N4868A booster module can be connected to the two differential output connectors of an E4866A 10.8 Gbit/s data generator module.

NOTE The cables have to have precisely the same delays. Because the signal has a period in the range of 100 ps, even a small difference would cause performance degradation in differential mode.

The standard booster module has two amplifiers, and you can specify their operation mode: differential (this is the default) or single-ended.

In single-ended mode, you can operate each amplifier separately. For example, you can connect two 10.8 Gbit/s generators—both set to single-ended operation—and amplify two separate signals.

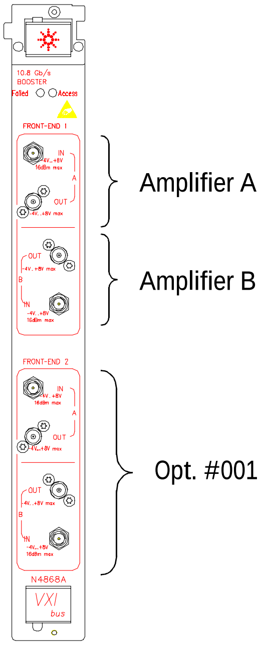


Figure 14 10.8 Gbit / s Booster Module

The figure above shows an N4868A module with option #001. It has two frontends and can thus be connected in differential mode to two E4866A 10.8 Gbit/s data generator modules.

For information on how to set the module mode and parameters, see *“How to Set Up N4868A Booster Channels” on page 255*.

Frontends

The available frontends include data generator and data analyzer frontends.

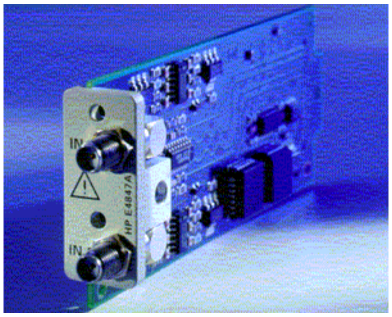


Figure 15 **Frontend With Two Connectors**

Note that the frontends have built-in protection circuits which automatically disconnect a frontend if an attempt is made to operate the frontend under intolerable conditions.

NOTE If this happens, the user interface is neither informed nor updated. In case of a problem, you should therefore always inspect the green LEDs above the frontend connectors. They clearly indicate the physical connection status.

Once the termination conditions have been corrected, the Connectors On/Off button of the toolbar can be used to re-establish the connection.

TIP Remote programs can query the connector status.

Generator Frontends

Generator Frontends for E4832A Modules

The generator frontends for the E4832A module are:

- E4838A, 675 MHz, differential output, low voltage amplitude/offset and variable slopes generator
- E4843A, 675 MHz, NRZ/RZ, differential output frontend

Generator Frontends for E4861A Modules

The generator frontends for the E4861A module are:

- E4862A, 2.7 GHz, differential generator frontend
- E4864A, 1.65 GHz, differential generator frontend

NOTE Your system may include older generator frontends, marked for frequencies of 2.67 GHz, 1,35 GHz, 1.33 GHz, or 667 MHz. If this is the case, the achievable data rates are limited by the older frontends.

Generator Frontends for E4861B Modules

The generator frontend for the E4861B module is:

- E4862B, 3.35 GHz, differential output frontend with variable delay and variable signal crossing

For details please refer to the *Agilent 81250 Technical Specifications*.

Analyzer Frontends

Analyzer Frontends for E4832A Modules

The analyzer frontend for the E4832A module is:

- E4835A, 675 MSa/s, differential/single-ended input, high sensitivity analyzer

The E4835A frontends are always installed in pairs. Two E4835A frontends share a common memory plug-in and sequencer.

Analyzer Frontends for E4861A Modules

The analyzer frontends for the E4861A module are:

- E4863A, 2.7 Gsa/s, differential/single-ended input frontend
- E4865A, 1.65 Gsa/s, differential/single-ended input frontend

Analyzer Frontend for E4861B Modules

The analyzer frontend for the E4861B module is:

- E4863B, 3.35 Gsa/s, differential /single-ended input frontend

The analyzer frontends for data rates above 675 Mbit/s have an AUX OUT connector which provides the input signal as interpreted by the input comparator—either low or high.

This output can for example be used to synchronize a pure analyzing system. If the DUT (such as a deserializer or demultiplexer) generates a recovered clock, this clock signal may be fed into the frontend, and the AUX OUT signal can be used to provide the clock to the analyzing system via the EXT INPUT connector of the clock module.

Note that the AUX OUT connector has an internal impedance of 50 Ω which must be met by the receiver in order to achieve the specified characteristics. A termination voltage between 0 V and –2 V may be used. If these requirements are not met, the output is disabled.

See also *"How Do I Use the AUX OUT of Analyzer Frontends?" on page 412*.

NOTE Your system may include older analyzer frontends, marked for frequencies of 2.67 GHz, 1.35 GHz, 1.33 GHz, or 667 MHz. If this is the case, the achievable data rates are limited by the older frontends.

For details please refer to the *Agilent 81250 Technical Specifications*.

Trigger Pod

The Agilent 81200 Trigger Pod is an option of the E4805B or E4808A clock module. It can be used to detect external events and to react on them.

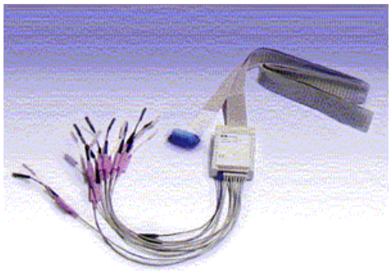


Figure 16 Agilent 81200 Trigger Pod

The ribbon cable has to be plugged into the master clock module.

The Trigger Pod has 8 TTL compatible input lines (input threshold 1.5 V). The input lines are terminated by 4.7 kΩ pull-up resistors to +5 V.

Data acquisition and hence event recognition is triggered by the internal sequencer clock. The sequencer clock frequency is:

```
Sequ. clock = System clock frequency / Segment resolution
```

The maximum sequencer clock frequency is 42.188 MHz, corresponding to a period of 23.7 ns.

For technical details please refer to the *Agilent 81250 Technical Specifications*.

The input lines can be used to detect single, asynchronous events. If certain patterns (bit combinations) are to be detected, it is recommended to synchronize the incoming data with the system. This can be done by generating a clock signal at the TRIGGER OUTPUT of the clock module and applying that clock to the event source (see also *"How to Set the Characteristics of the Trigger Output" on page 192*).

Identification of Hardware Resources

The hardware resources pool is comparable to a traditional instrument. Here the instrument is seen as a collection of modules, which provide signal connectors.

An Agilent 81250 Parallel Bit Error Ratio Tester can consist of multiple **clockgroups**. Each clockgroup consists of **modules** which in turn have **connectors**.

The following figure illustrates the numbering system used to address the modules and connectors.

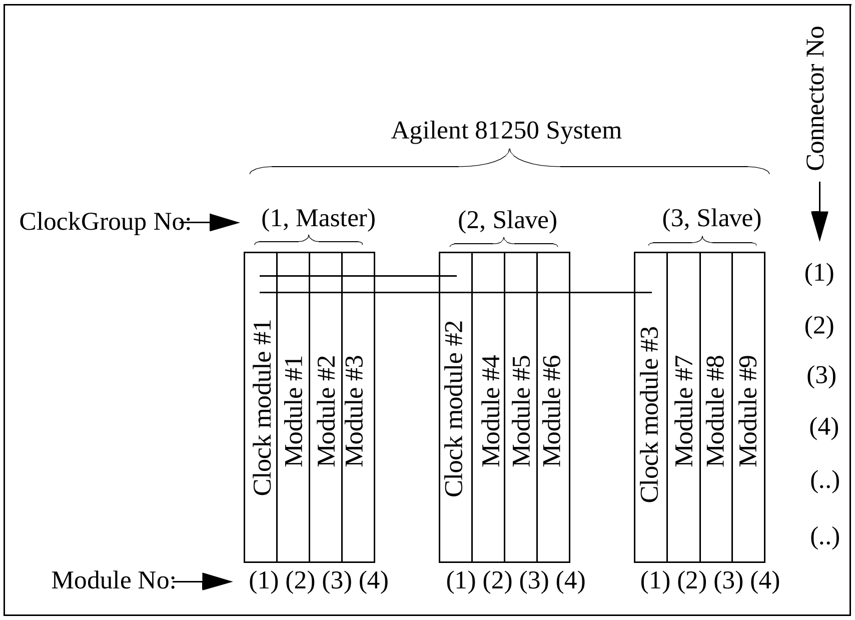


Figure 17 Numbering of Modules and Connectors Within One System

The identification of a generator or analyzer channel is:

Cx My Cz (ClockgroupNumber – ModuleNumber – ConnectorNumber), such as C1 M3 C5.

NOTE An Agilent 81250 Parallel Bit Error Ratio Tester can also house several systems. A **system** requires:

- one independent master clock module, and
- one or more data generator/analyzer modules.

Summary of Hardware-Related Terms

Module: One of the following:

- E4805B Central Clock Module
- E4808A Central Clock Module
- E4832A 675 Mbit/s Data Generator/Analyzer Module
- E4861A 2.7 Gbit/s Data Generator/Analyzer Module
- E4861B 3.35 Gbit/s Data Generator/Analyzer Module
- E4810A 3.35 Gbit/s optical-electrical Data Generator Module
- E4811A 3.35 Gbit/s optical-electrical Data Analyzer Module
- E4866A 10.8 Gbit/s Data Generator Module
- E4867A 10.8 Gbit/s Data Analyzer Module
- N4868A 10.8 Gbit/s Booster Module

Frontend: Generator or analyzer plug-in of a data generator/analyzer module.

Connector: An output or input connector of a frontend. Differential connectors are treated as one connector.

Channel: The circuitry behind a connector, which includes data generating or analyzing capabilities, data memory, frequency multiplexing and so on.

Trigger Pod: An option of the master clock module for detecting external events.

Master clock module: The clock module that controls clockgroup #1 of a system. It can additionally control up to two slave clock modules (clockgroup #2 and #3).

Applications above 2.7 Gbit/s require the E4808A High Performance Clock Module.

Clockgroup: The sum of modules connected to a single clock module.

ClockgroupNumber: Identifies the clock master (= 1) and up to two slaves (2 and 3).

ModuleNumber: Identifies the module within a clockgroup (1 to 11).

ConnectorNumber: Identifies the connector of a module and is counted from module top to bottom (1 to 4). Differential connectors are counted as one connector.

Operating Principles

The software of the Agilent 81250 Parallel Bit Error Ratio Tester is based on two concepts:

- The idea of **virtual systems**
- The idea of using **models** of the real world—a model of the present instrument configuration and a model of the DUT

This section explains the interdependencies and terms. See:

- *"Virtual Systems" on page 58*—one ParBERT system can occupy up to three mainframes; also, multiple independent systems can be installed in one mainframe and operated by one controller
- *"Hardware and Setup Models" on page 60*—an introduction to the Connection Editor that mirrors the physical world
- *"Software Structure" on page 62*—an overview of the manifold ways that can be used for controlling a ParBERT system

Virtual Systems

A single Agilent 81250 system can contain up to three clock modules in a master-slave configuration. Such a configuration uses expander frames and has up to three clockgroups.

On the other hand, it is also possible that one mainframe houses several Agilent 81250 system*s*. They consist of *independent* clock modules with associated data generator/analyzer modules. Such a configuration makes it possible to test a device under asynchronous conditions using independent clock pulses.

The concept of the Agilent 81250 Parallel Bit Error Ratio Tester is to create so-called virtual systems from the system's present hardware resources (clock modules, data modules, generator and analyzer frontends).

The basic (default) system is called DSRA (DSR = digital stimulus and response, system A).

If the hardware comprises several independent clock modules—not connected as slaves—then additional systems are available. By default, they have ascending names, such as DSRB, DSRC, and so on.

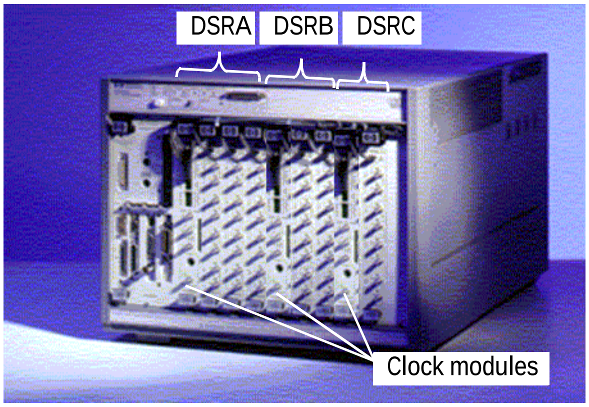


Figure 18 **Virtual Systems in One Mainframe**

Separate systems are widely used for testing multiplexers or demultiplexers. Such devices generally require different clock frequencies at the generating and analyzing sides. MUX/DEMUX tests can be performed by one system, if the ratio of these frequencies is 2^n, such as 2, 4, 8, 16, and so on. If it is not, separate data generating and data analyzing systems will do the job.

If the hardware has been changed by adding or removing modules or frontends, new virtual systems can be automatically created with the Agilent 81250 Configuration Tool (see also *"How to Set the Operating Mode" on page 135*).

The systems are identified in the file *dvtsys.txt*. By editing the *dvtsys.txt* file, they can be renamed. Their configurations are automatically detected and stored in the *dvtits.txt* configuration file.

NOTE The user interface and remote control commands enable you to load and operate any of the configured subsystems.

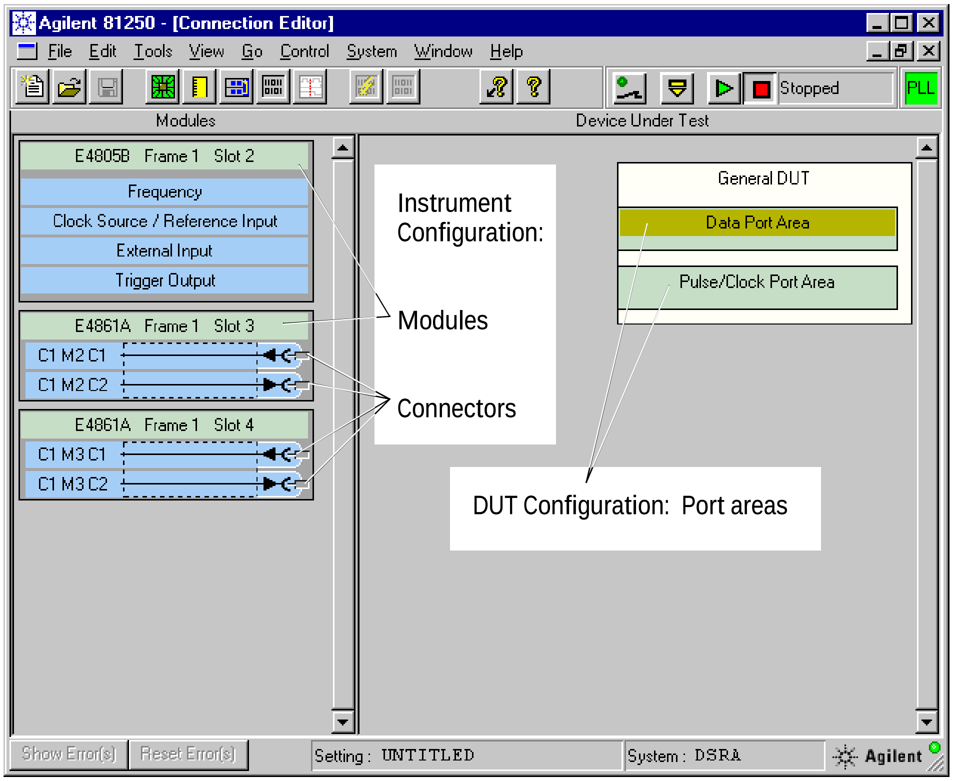
You can operate several virtual systems in parallel by starting the user interface more than once. Every user interface indicates the chosen system in the bottom line of the main window.

If the tester is operated remotely via SCPI commands, the system names are used to construct the **handles** for identifying the respective system.

Hardware and Setup Models

On power up, the system automatically checks and identifies the available modules and frontends. It creates an image of the system configuration and displays this image in the Connection Editor window.

The image of the DUT needs to be created. This can be done manually with the Connection Editor or by loading a stored setting.



Setting : UNTITLED

System : DSRA

Figure 19 Connection Editor Window

System Configuration

The system configuration at the left-hand side of the Connection Editor window identifies all the available modules, frontends (there are generator and analyzer frontends), and frontend connectors.

Refer to *“Frontends” on page 53* for the available frontends.

A **connector** represents an output or input connector of the module. Differential connectors are treated as one connector.

A **channel** represents the circuitry behind a connector. It is identified by Cx My Cz (ClockgroupNumber - ModuleNumber – ConnectorNumber).

DUT Configuration

The image of the DUT is constructed from a template displayed on the right-hand side of the Connection Editor window. The Agilent 81250 system includes a general DUT template.

The general DUT template provides two types of **ports**:

- Data ports

  Data ports provide the ability to define data to be sent or analyzed. Data is handled in the form of segments. Two major data segment types are available: memory-based or PRBS/PRWS. Sequences of segments can be repeated or started upon events.

  A data port is usually an input or output bus, characterized by one common clock frequency.

- Pulse port

  Like traditional pulse generators, this port provides an easy way to have a pulse generated without the need to set up any data. The terminals of a pulse port can receive different clock signals.

Ports can be added for every group of pins with the same or similar behavior.

Ports consist of **terminals**. These are the DUT pins that must be physically connected to the connectors of the system's generator or analyzer channels.

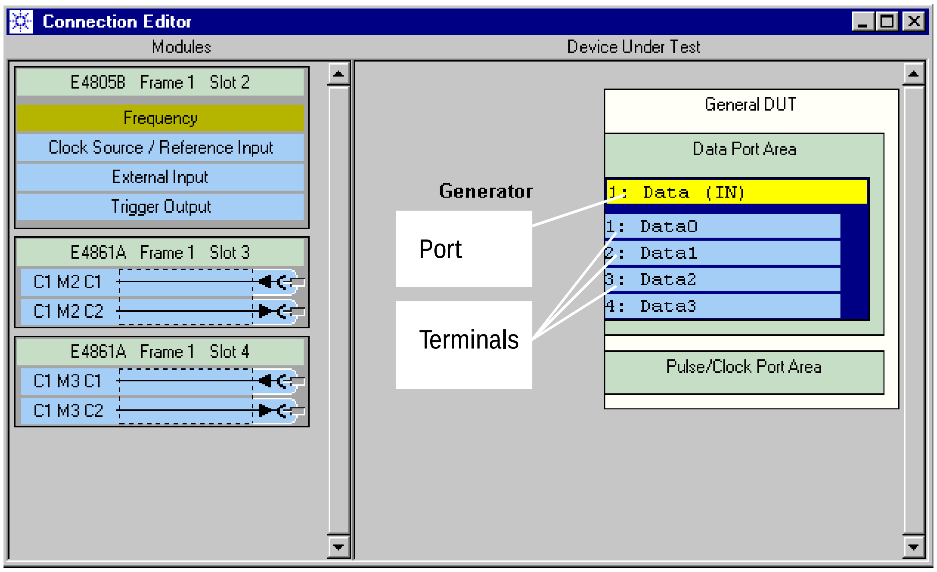


Figure 20 Display of Ports and Terminals

After the virtual DUT has been created and connected to the virtual system, signal parameters can be set, such as signal timings, voltages, signal termination characteristics, pulse delay, pulse width, and so on. Signal parameters may be set up globally for a whole port or individually for single terminals.

More complex signals can be produced by **digital addition** of two or four output channels. This allows to generate real-world signals with pulse displacement or width variation.

Another feature is **analog channel addition** which allows to generate signals with glitches, distorted transitions, and multiple levels.

Settings

The complete setup for a DUT including all parameters is called **setting**. Settings can be saved in the system's database. It is also possible to export/import a setting as a text file, either manually or under remote control.

Every saved setting can be reloaded at any time.

A setting also contains references to the signal patterns used for the test. These patterns are stored as segments. If a setting is exported to or imported from another system, the required segments have to be exported/imported as well. Therefore, all the segments required by a setting can be stored in a "local" segment pool which is associated with and only accessible from the setting.

NOTE Settings include the system configuration. Stored settings can therefore not be used on systems that do not provide the required hardware configuration.

Software Structure

The Agilent 81250 Parallel Bit Error Ratio Tester can be controlled from a variety of interfaces:

- Graphical user interface

  The graphical user interface allows you to set up and execute device tests interactively using the keyboard and mouse.

- GPIB interface

  The GPIB interface allows remote control of the Agilent 81250 system via the General Purpose Instrument Bus interface.

- User-written programs

  The Agilent 81250 software provides application programming interfaces (APIs) for a couple of programming environments, such as Microsoft's C/C++, VisualBasic, Agilent's VEE, National Instruments' LabVIEW, and others.

- Local area network (LAN)

  The graphical user interface or user-written programs can be run on any Windows NT or Windows 2000 workstation. Once the Agilent 81250 ParBERT is connected to the LAN, its firmware server can be operated from any workstation via the LAN.

The software is based on a client-server architecture. All the interfaces communicate with the module firmware through one and the same software component—the **firmware server**.

The system can be remote-controlled via plug and play functions as illustrated below:

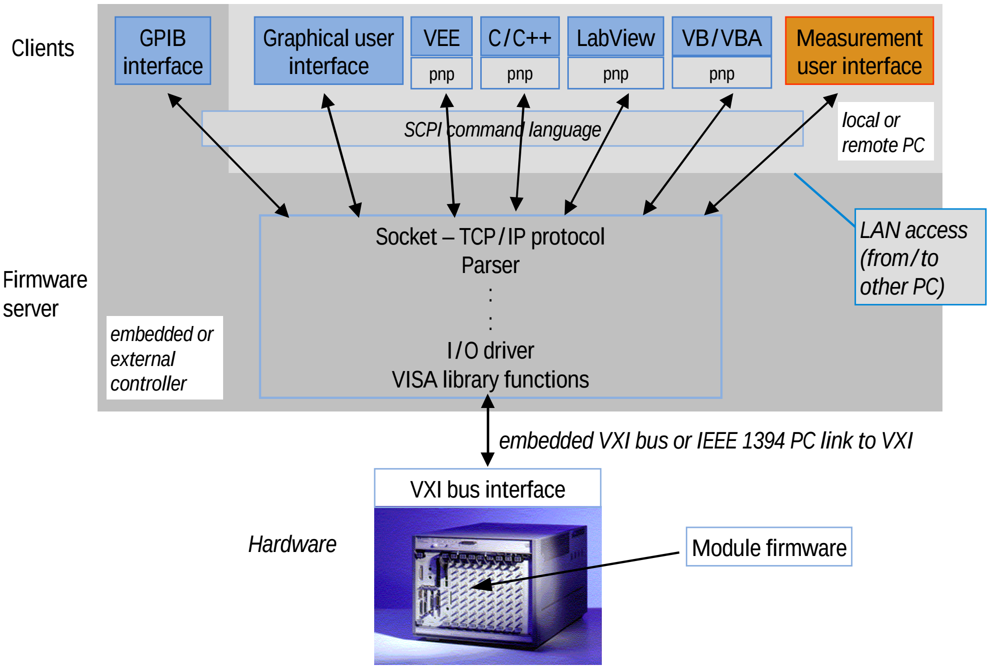


Figure 21 Remote Control Options

Plug and play functions

The Agilent 81250 system software includes the Agilent 81200 plug and play (pnp) driver. The pnp functions can be called from a variety of programming environments, such as VEE, C/C++, VisualBasic, or others. VB/VBA, for example, is supported by Microsoft Excel or even Word. The Agilent 81200 pnp driver performs the protocol conversion to the SCPI command language.

GPIB access

The system can also be controlled via the General Purpose Instrument Bus interface. This interface uses the SCPI command language.

The GPIB interface has to run on the same PC as the firmware server. All the other interfaces can also run on any suitable workstation of the LAN.

LAN access

The built-in controller has a Local Area Network interface that can be accessed from any workstation. An 81250 user interface or a user-written program running on such a workstation can communicate with the firmware server running on the Agilent 81250 Parallel Bit Error Ratio Tester.

External controller

The built-in controller can be replaced by an external PC using the IEEE 1394 PC link to VXI. In this case the firmware server runs on that PC.

Measurement Software

The Agilent 81250 Measurement Software has been developed using the ActiveX concept of common, reusable software objects, generally called **ActiveX controls**. There are no SCPI commands that allow to control the measurements. But they can be embedded.

As the measurements are implemented as ActiveX controls—the measurement framework is just a “container”—the measurements can be easily embedded into any program development environment that supports the component object model, such as Microsoft Excel or Word.

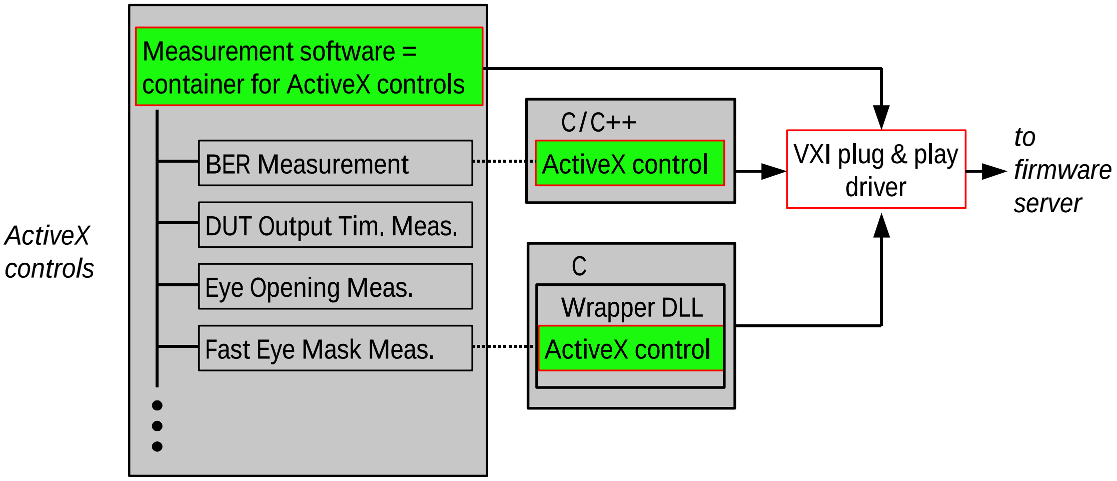


Figure 22 Measurement ActiveX Components

For compatibility reasons, the software provides also a Wrapper DLL. This library allows you to embed ActiveX controls into pure C programs.

NOTE It is recommended to use the remote access generally via the plug and play functions. For the Agilent 81250 Measurement Software, no SCPI commands are provided.

SCPI command language All interfaces finally use an ASCII-driven serial interface protocol to communicate with the firmware server. The interface protocol is based on the SCPI command language.

In remote-controlled operation, each virtual subsystem needs an individual handle so that the software knows which subsystem should receive the command for executing it. All SCPI commands and queries have to start with the subsystem's specific handle.

Summary of Setup-Related Terms

Port: A group of DUT input or output pins with identical or similar properties, such as a data or address bus.

Data port: A DUT port that receives or returns digital data.

Pulse port: A DUT port that receives parametric signals such as a clock pulse.

Terminal: A signal line assigned to a port (a DUT pin).

Setting: The complete setup for a DUT including all parameters and references to the test patterns (segments).

Handle: The identification of a subsystem, such as DSRA.

API: Application Programming Interface.

SCPI: Standard Command language for Programmable Instruments.

PNP: Plug and Play peripheral driver for VXI components.

BIOS: Basic I/O System—the microprocessor programs loaded into the modules' EEPROMs.

VISA: Virtual Instruments Software Architecture—a common standard of functional calls for controlling VXI-based instruments.

Agilent 81200 pnp driver: A plug and play driver for the 81200 platform based on the VISA standard.

ActiveX controls: Reusable software objects that can be pasted into many programming environments.

Timing Principles

Depending on the modules and frontends, it is possible to test devices at frequencies up to 10.8 GHz. ParBERT 43G systems support data rates up to 43.2 Gbit/s. Multiple frequencies and memory resolutions are achieved by multiplying the clock frequency.

See:

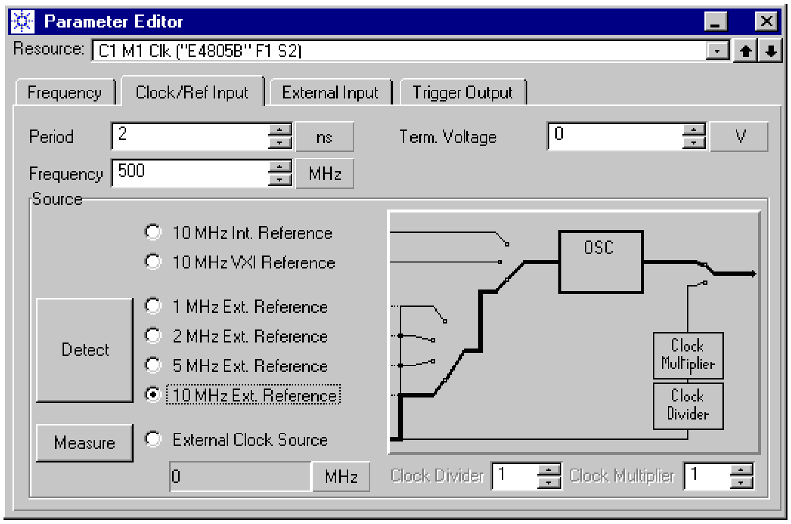
- *“Choice of Clock Sources” on page 66*—the built-in clock module can generate the system clock or lock to an external clock source
- *“Frequency Multiplier and Segment Resolution” on page 68*—here, you find the explanation of how the system generates data streams and how the chosen frequency affects the available memory depth
- *“Adding 675 Mbit/s Generator Signals” on page 73*—an overview of the available features for generating signals that are not uniform with respect to timing or amplitude
- *“Signal Delay Compensation” on page 75*—an introduction to the principles of zero adjust, cable delay, and propagation delay compensation
- *“Trigger-Controlled Start and Stop” on page 76*—basic information on how tests can be started and stopped by an external signal

Choice of Clock Sources

The Agilent 81250 system has a built-in 10 MHz reference oscillator, but can also be locked to an external clock source.

The external clock can be used to substitute the built-in reference. It can also be used to drive the system directly. It can be connected to the CLOCK/REF INPUT of the master clock module and set up with the Parameter Editor.

The Parameter Editor indicates the chosen clock source path as illustrated in the following figure.



Parameter Editor

Resource: C1 M1 Clk ("E4805B" F1 S2)

Period 2 ns

Term. Voltage 0 V

Frequency 500 MHz

Source

○ 10 MHz Int. Reference

○ 10 MHz VXI Reference

○ 1 MHz Ext. Reference

○ 2 MHz Ext. Reference

○ 5 MHz Ext. Reference

● 10 MHz Ext. Reference

○ External Clock Source

0 MHz

Clock Divider 1

Clock Multiplier 1

Figure 23 Clock / Reference Input Window

In this example, an external 10 MHz reference has been chosen.

You can see from the picture that the supported clock sources are grouped:

- Internal reference clock sources include the oscillator of the clock module and the VXI bus clock signal.
- External reference clock sources have to run at one of the supported clock frequencies.
- A completely independent external clock source can be applied, measured, and adjusted (multiplied/divided) for generating the system clock.

Independent from the clock source, the system can be started by an external signal applied to the EXT INPUT connector of the master clock module.

A system that contains only E4832A data generator/analyzer modules can also be stopped or gated (started and stopped) by an external signal.

For details see *“Setting Global System Parameters” on page 175.*

Frequency Multiplier and Segment Resolution

The internal data handling is based on words. The number of bits which are allocated to one word depends on the **segment resolution**.

Let us have a look at the architecture of a module and its frontends in order to understand the dependencies between

- the system clock frequency,
- the word length, and
- the available data memory,

The following figure illustrates the structure of a data generation channel of an E4861A module:

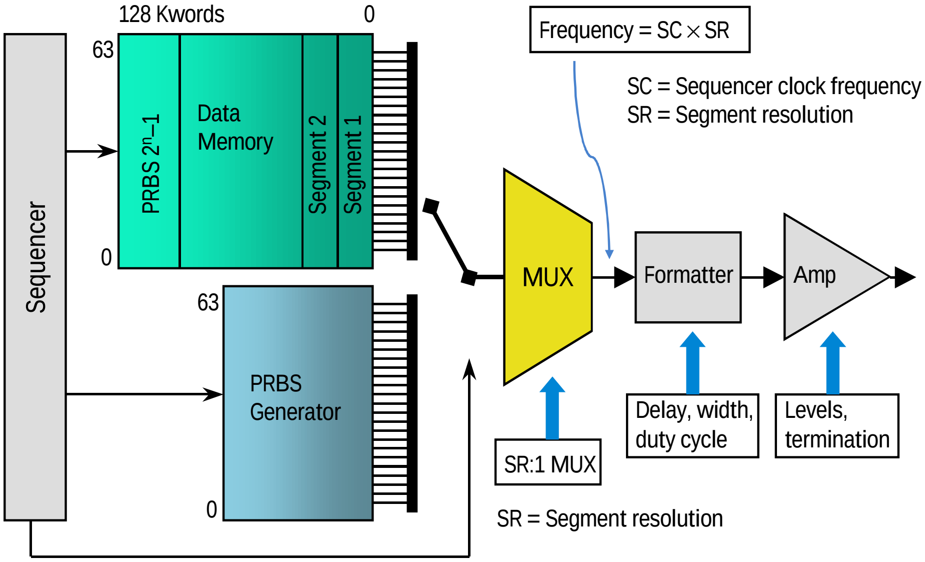


Figure 24 E4861A Hardware Architecture

Channel data memory

Memorized data such as data segments or distorted PRBS is downloaded from the system controller and kept in the channel memory. The memory of an E4861A channel can hold 128 Kwords of 64 bits. The memory of an E4832A channel can hold 128 Kwords of 16 bits. The memory of an E4866A generator or E4867A analyzer module can hold 128 Kwords of 256 bits.

PRBS generator

Pure PRBS/PRWS signals are generated by a hardware feedback shift register (see also *“Appendix B: PRBS/PRWS Data Segments” on page 425*).

Sequencer

Both memory and PRBS generator are controlled and clocked by the built-in sequencer. The sequencer clock–generated and distributed by the master clock module of the system–has a maximum frequency of 42.1875 MHz.

Multiplexer

Depending on the kind of data to be generated, either the memory or the PRBS generator is connected to a multiplexer. This multiplexer generates the data stream.

When you set a desired system clock frequency, then the sequencer clock is automatically adjusted so that the desired frequency is an integer multiple of the sequencer clock frequency.

This multiplexing factor determines the word length–the number of bits polled by the multiplexer–and hence the available memory capacity. It is called **segment resolution**.

NOTE There is one segment resolution with global scope. If the system generates multiple clock frequencies, then some ports or channels have individual segment resolutions.

Segment Resolution

The specified and generated data rate is:

```
System frequency = Sequencer clock frequency × Segment resolution
```

In case of a single frequency system, the segment resolution is the general segment resolution. In case of a system using multiple data rates, the segment resolution is the individual segment resolution of the channel.

If the segment resolution is lower than the maximum number of bits per word provided by a data generator/analyzer module, then the remaining bits are not used.

Segment resolution and memory depth

The available frequencies and resolutions depend on the type of module and its frontends.

The E4832A module provides 128 K of 16-bit words of memory for each channel. Depending on how many bits are used, this results in 128 Kbit to 2 Mbit of usable memory.

If 16 bits are allocated to a word, it is possible to have signals with data rates up to 675 MHz and 2 Mbit data memory.

Table 3 Word Length, Memory Depth, and Frequency Range of an E4832A Module

Segment resolution (bits)	Max. memory depth (bits)	System clock frequency (Mbit / s)
1	131,008	≤ 42.1875
2	262,016	≤ 84.375
4	524,032	≤ 168.750
8	1,048,064	≤ 337.500
16	2,097,152	≤ 675.000

The E4861A module has a memory capacity of up to 8 MB per channel. Its minimum segment resolution is 16 bits (see the table below):

Table 4 Word Length, Memory Depth, and Frequency Range of an E4861A Module

Segment resolution (bits)	Max. memory depth (bits)	System clock frequency (Mbit / s)
16	2,097,152	333.334 to 675.000
32	4,194,304	≤ 1,350.000
64	8,388,608	≤ 2,700.000

The E4861B module allows to use segment resolutions from 1 to 128 bits (see the table below):

Table 5 Word Length, Memory Depth, and Frequency Range of E4810A, E4811A, and E4861B Modules

Segment resolution (bits)	Max. memory depth (bits)	System clock frequency (Mbit / s)
1	131,072	20.834 to 41.666
2	262,144	≤ 82.333
4	524,288	≤ 166.666
8	1,048,576	≤ 333.333
16	2,097,152	≤ 666.666
32	4,194,304	≤ 1,333.333
64	8,388,608	≤ 2,700.000
128	16,777,216	≤ 3,350.000

Increasing the virtual memory

The desired system clock frequency determines the minimum segment resolution. As long as the segment resolution is less than 16 (or 64 for E4861A modules), it is possible to choose a higher general segment resolution. This increases the usable memory and simultaneously decreases the sequencer clock frequency.

NOTE For E4832A modules, the delay vernier and the functions for automatic analyzer sampling delay adjustment require that the minimum segment resolution of the desired system clock frequency is used.

The E4866A and E4867A modules have only one segment resolution:

Table 6 Word Length, Memory Depth, and Frequency Range of E4866A / E4867A Modules

Segment resolution (bits)	Max. memory depth (bits)	System clock frequency (Gbit / s)
256	33,554,432	9.5 to 10.8

Block length

The streams of generated or expected data are organized in blocks. Every block must contain a whole number of words. Therefore, the length of these blocks has to be a multiple of the segment resolution.

Multiple Frequencies and the Frequency Multiplier

The general segment resolution is a global system parameter. It determines the sequencer clock frequency.

FM factor and FMR

The data rate of individual channels or ports can be changed by changing the **frequency multiplier factor (FM factor)**.

The chosen FM factor for a port or channel determines not only its frequency but also the individual segment resolution of this port or channel. The individual segment resolution is

```
SR = General segment resolution × FM factor
```

Limitations:

```
For E4832A modules: 1 ≤ SR ≤ 16
For E4861A modules: 16 ≤ SR ≤ 64
For E4861B modules: 1 ≤ SR ≤ 128
```

The available multiplying factors are expressed by the **frequency multiplier range (FMR)**.

The system calculates the FMR of a channel automatically. The selection box provided for changing the FM factor offers only valid factors.

E4832A-Example

The following example refers to the E4832A module.

If the desired clock rate is 100 MHz, then the minimum segment resolution is 4, which leads to 512 Kbit memory depth and a frequency multiplier range of 1/4, 1/2, 1, 2, 4. That means, an individual channel can run at 25 MHz, 50 MHz, 100 MHz, 200 MHz, or 400 MHz.

Other possible segment resolutions for this clock rate are:

- 8, which leads to 1 Mbit memory depth and FMR = 1/8, 1/4, 1/2, 1, 2.
- 16, which leads to 2 Mbit memory depth, FMR = 1/16, 1/8, 1/4, 1/2, or 1.

The relations are shown in the table below:

Table 7 Matrix of Segment Resolution, FMR, Memory Depth and Clock Frequency

Segment Resolution	Frequency Multiplier Range[a]	Memory Depth[b]	Channel Clock Frequency
1 bit (=1)	1, 2, 4, 8, 16	128 Kbit	≤ 42.1875 MHz
2 bits (=2)	1/2, 1, 2, 4, 8	256 Kbit	≤ 84.375 MHz
4 bits (=4)	1/4, 1/2, 1, 2, 4	512 Kbit	≤ 168.75 MHz
8 bits (=8)	1/8, 1/4, 1/2, 1, 2	1 Mbit	≤ 337.5 MHz
16 bits (=16)	1/16, 1/8, 1/4, 1/2, 1	2 Mbit	≤ 675 MHz

[a] This is the range of multiples and fractions that can be used at individual connectors. If you have most of your signals at 40 MHz and your pattern lengths are less then 64 Kbit, then you can choose segment resolution 1. You have the chance to set individual connectors to a multiple of this general setting. For example, selecting 16 as the multiply factor for a connector gives you 1 Mbit memory depth and 640 MHz with a segment resolution of 16.

[b] Subtract 32 x segment resolution, as this memory space is occupied by a 2^5-1 PRxS and the sequencing initialization.

If, for example, most of the signals are at 200 MHz, then the available segment resolutions are either 8 or 16.

If you have chosen 8 as the general segment resolution, then each data port and each terminal of a pulse port can be set individually to frequencies of 1/8, 1/4, 1/2, 1 or 2 times the system clock frequency.

NOTE If the frequency multiplying factor is changed for individual ports or channels, then the frequency, word length, and memory depth also change for these connectors.

Mixing Low and High Speed Modules

The data generator/analyzer modules E4832A and E4861A have one frequency range in common: 333.334 MHz to 675 MHz.

If a system contains a mixture of E4832A and E4861A modules and the system clock frequency is set to a value within this range, then a general segment resolution of 16 is automatically set and used on all channels.

If a lower system clock frequency is set, then the frequencies of the high-speed channels are automatically multiplied. This is done by setting the frequency multiplier to a value higher than one.

The FM factor is chosen such that the high-speed channels operate with their minimum segment resolution, which is 16 for an E4861A module.

Example

If a system clock frequency of 125 MHz has been chosen, then the minimum segment resolution for the E4832A modules is 4 (see also the table *"Word Length, Memory Depth, and Frequency Range of an E4832A Module" on page 70*). This is automatically set as the general segment resolution.

The frequency multipliers of the E4861A module channels are set to 4, resulting in individual segment resolutions of 16 and a channel frequency of 500 MHz.

For details see *"How to Use Multiple Frequencies" on page 182*.

Adding 675 Mbit / s Generator Signals

The outputs of 675 Mbit/s generator frontends can be digitally added. This allows to generate signals with varying pulse widths. It allows also to generate signals with data rates above 675 Mbit/s even with low-speed frontends.

The output of an E4838A generator frontend can also be added in analog mode to the output of the generator above. This allows to generate signals with spikes or multiple levels.

Digital Channel Addition

Two or four channels can be digitally added. The digital channel addition is an XOR addition (exclusive OR or modulo 2 addition). The addition takes place before levels are applied to the signals.

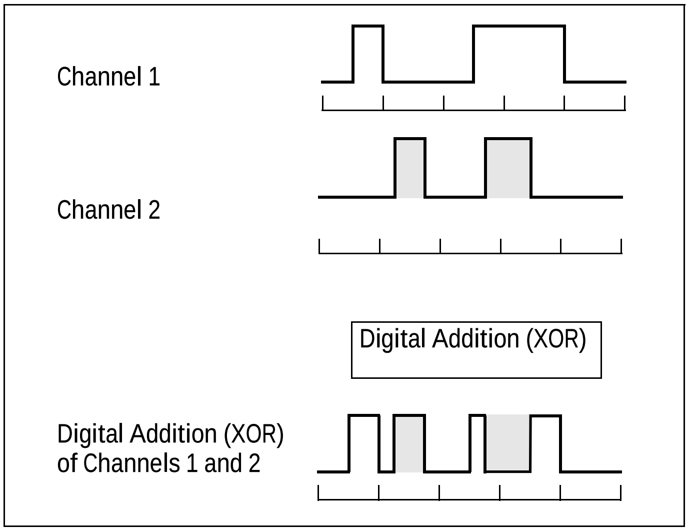


Figure 25 **Digital Channel Addition**

For details see *"How to Combine Generator Channels" on page 251*.

Analog Channel Addition

Analog channel addition works as illustrated in the figure below:

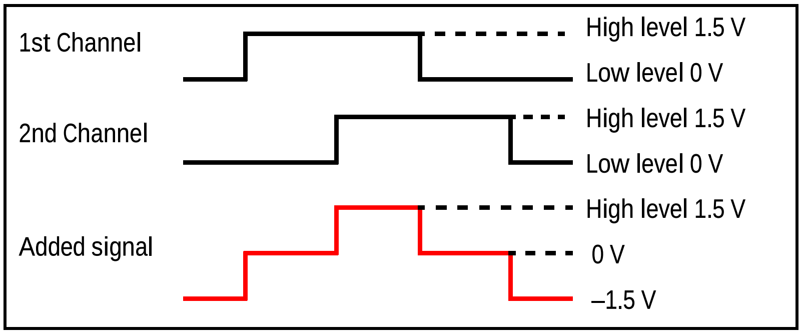


Figure 26 **Analog Channel Addition**

Two output voltages are added to form a signal with three voltage steps.

Both channels can have different timing parameters, such as frequency, pulse width and delay. High level and expected load are determined by the channel that holds the connector.

For details see *"How to Add Channels in Analog Mode" on page 240*.

Signal Delay Compensation

Several features support the timely alignment of generated and captured signals.

Zero Adjust, Cable, and Propagation Delay Compensation

With the Agilent 81250 system, it is possible to align the outputs and inputs of newly installed frontends or new modules.

To assure that all generator output signals are applied at the same time either at the DUT board or even at the DUT input pins, it is possible to perform a cable delay and propagation delay compensation.

The various kinds of signal delay are illustrated in the figure below.

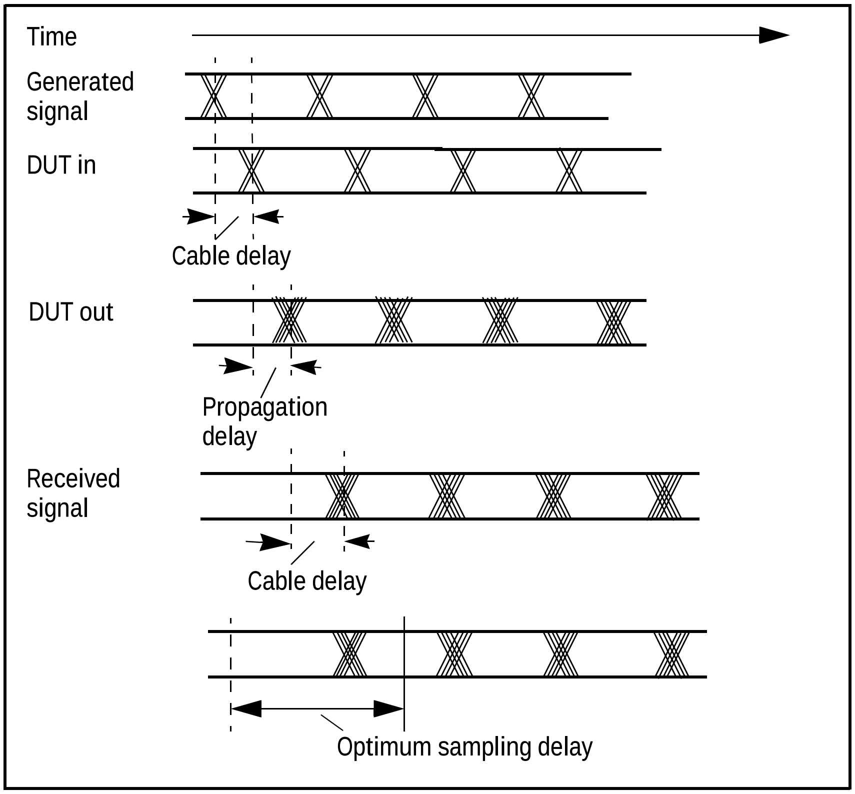


Figure 27 **Kinds of Signal Delay**

The procedures are menu driven and semi-automatic. For details see *"How to Compensate for Internal and External Delays" on page 364.*

Negative Delay

Apart from aligning signals it may be interesting and important to have some signals applied in advance to other signals.

Therefore the Agilent 81250 system provides the option to set a general time offset for all connectors, so that individual ones can be set to negative delays, and hence start earlier than others.

The delay offset feature can be used in setup and hold time measurements. For details see *"How to Set the General System Frequency" on page 178.*

Trigger-Controlled Start and Stop

The EXT INPUT of the master clock module can be used to start the timing system of the Agilent 81250. The state of this input is sampled once every system period. If the system does not contain an E4861A module, this input can also be used to stop the system.

System period The system period is normally derived from the built-in PLL. However, if the CLOCK/REF INPUT provides an external clock source, this signal defines the system period.

When using the EXT INPUT without an external clock source at the CLOCK/REF INPUT, the start/stop signal must be applied for a time greater than the system period.

When using the EXT INPUT with an external clock source at the CLOCK/REF INPUT, setup and hold times must be considered to achieve a predictable timing.

NOTE Setup and hold time violations may influence the Agilent 81250 system only in the aspect that the system will generate consistent relative timings for data generation and data capture; the absolute timing, related to the external input, may vary by about ±1 system periods.

Trigger-Controlled Start Starting the system via the EXT INPUT has no restrictions. All internal pipelines are prefilled so that the first signal comes out after

```
±1 system periods + 45 ns + output delay
```

NOTE If trigger-controlled start is used in combination with an external source clock, the external clock must be stable and applied at least 100 ms prior to the start pulse. This is about the time required by the built-in PLL to lock safely on the external clock.

Trigger-Controlled Stop Trigger-controlled stop or trigger-controlled gating via the EXT INPUT is only available if the system contains exclusively E4832A modules.

When stopping the system via the EXT INPUT some restrictions apply.

The timing system is stopped immediately, even if the period and delay of a bit is not complete. The consequence is, that the word at the output might not be aligned during such a stop. After a restart (in gated mode), the bits are realigned.

When stopping the system, ensure that the output word is stable for a longer time. Use a PAUSE segment for that purpose. The pause should have a duration of

```
System period × Frequency multiplier × 32 + Maximum delay
```

where the maximum delay is the maximum delay of all involved channels. Details are documented in the *Agilent 81250 Technical Specifications*.

Summary of Timing-Related Terms

Segment resolution: The length of a data word. Range: 1 to 16 bits for E4832A modules, 16, 32, or 64 bits for E4861A modules. The minimum segment resolution depends on the chosen system clock frequency.

FM factor: Frequency Multiplier factor. The individual factor by which a channel or port frequency differs from the system clock frequency. Choices are restricted by the FMR.

FMR: Frequency Multiplier Range. The available factors for multiplying the system clock frequency. The actual range depends on the segment resolution and the type of module.

Block: A portion of a test sequence which references segments, that define generated or expected data. A block refers to all data ports. Its length has to be a multiple of the segment resolution.

Digital channel addition: Exclusive OR (modulo 2) addition of two or four signal generator channels of one 675 Gbit/s data module.

Analog channel addition: Voltage addition of two signal generator channels of one 675 Gbit/s data module. Requires at least one E4838A frontend.

EXT INPUT: A connector of the clock module. It allows to start/stop the system by an external signal.

CLOCK / REF INPUT: A connector of the clock module. It allows you to connect an external clock source.

Data Generation Principles

Once the general signal parameters have been set, the patterns of the data to be generated or expected can be defined.

See:

- *"Emulate Real Pattern and Waveform Conditions" on page 78*—an introduction to how data patterns to be generated or expected are treated
- *"Data Sequences" on page 79*—the general explanation of a data sequence that specifies for all data ports the data to be generated and expected
- *"Data Blocks" on page 80*—the explanation of the blocks that form a sequence
- *"Data Segments" on page 81*—an explanation of the data segments that are referenced by the sequence blocks
- *"Properties of Real Segments" on page 82*—describes the differences between memory-based and algorithmic (PRBS/PRWS) data segments
- *"Loops" on page 84*—an overview of how single or multiple blocks of a sequence can be repeated
- *"Hardware Dependencies" on page 85*—summarizes the hardware-dependent loop limits

Emulate Real Pattern and Waveform Conditions

Data patterns can be stored in the system database and output as part of a sequence with or without algorithmic data. A PRBS, for example, is algorithmic data.

Data segments

Data patterns for the signals sourced to or expected from the DUT can easily be set up in terms of data **segments** that span across several output or input connectors of the Agilent 81250 system.

Captured data or data produced by a simulation can be imported as an ASCII text file.

The Agilent 81250 system can be used to stimulate communication devices using its sequencing capability.

Packets or cells consisting of payload and control data can be produced by creating control segments, and using a PRBS segment for the payload. Cell/packet sizes can be varied and control segments can be stored in the database and used in any number of different packets. A PRBS pattern may be used as the payload to test error rates. Intermittent data with long dead-times between bursts can easily be produced using the pause segment.

For testing multiplexers/demultiplexers, it is possible to set up PRWS data and compare segments. Also, it is possible to run different ports at different frequencies.

Data sequence

The overall stream of generated and expected data is called a **sequence**. The Sequence Editor defines the structure of the data streams sent to or expected by the Agilent 81250 system.

NOTE

A sequence includes all data ports of the device under test. It does not include pulse ports.

Data Sequences

A sequence specifies which data segments are generated or expected, on which ports, and in which chronological order.

For details, see *“Creating the Stream of Generated and Expected Data” on page 265*.

Data blocks

A sequence consists of **blocks**. Loops are also part of a sequence.

Sequences are independent of data. This is achieved by defining the data to be generated or expected with the Segment Editor, or externally as vectors in a text file, and referencing these segments in the sequence blocks individually for each data port.

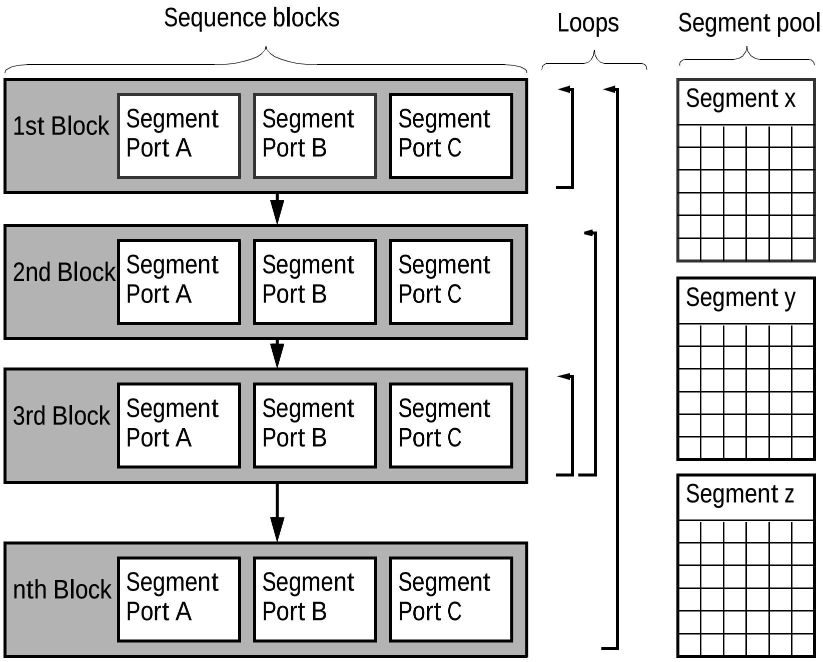


Figure 28 **Sequence and Segments**

Every block can reference an arbitrary segment for every data port.

Data Blocks

Blocks are portions of the sequence. A block spans across all data ports. For details see *"Contents of the Detail Mode Sequence Editor Window" on page 287*.

Every block references a **segment** for each DUT data port (not pulse ports). The segments contain the patterns of generated and expected data.

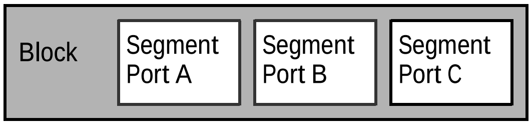


Figure 29 **Block Structure**

NOTE Segments are not included into, but only referenced by the blocks.

The length of a block must be a multiple of the segment resolution.

Single, several, or all blocks may be repeated a specified number of times or perpetually.

Such loops have an impact on the minimum block length and the allowed number of blocks (see *"Hardware Dependencies" on page 85*).

Trigger pulses can be specified to be generated at the beginning of a block, and output by the TRIGGER OUTPUT connector of the master clock module. If certain **events** have been detected while a block is executed, actions can be performed immediately, or at the end of the block. See *"Usage of Events" on page 105*.

A single block can also be used for synchronizing the analyzer frontends with the incoming data stream. See *"Principles of Analyzer Sampling Point Adjustment" on page 87*.

Data Segments

Segments can be freely created. A segment has a width and a length. The width defines the number of parallel signal lines (traces). The length defines the number of data words (vectors). The length of a segment must not remain under the length of the block into which it is going to be inserted.

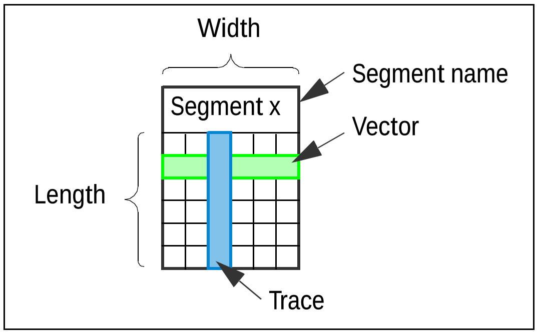


Figure 30 Segment Structure

Segments can be created separately or while editing the sequence. A segment may be larger than the block that references it.

If an existing segment is referenced by a block, the width of the port and the length of the block are automatically considered. Segment data that does not fit into the current block is ignored.

Segment Types

There are real segments and pseudo segments.

- Real segments contain either free programmable memorized data, or the specification of PRBS or PRWS data.
- Pseudo segments are commands, such as Pause, Don't care, Expect 0, Acquire, and so on. A pseudo segment consumes only one word of channel memory if it is used on all channels of a module.

New segments are created with the Segment Editor or can be imported from vector-formatted text files.

Segment Pools

Segments are stored in a segment pool, which is part of the system database. There is one segment pool with global scope and one segment pool per setting with local scope.

- Segments in the local segment pools can only be accessed if the appropriate setting is loaded.
- Segments in the global segment pool can be accessed from any setting.

Using the “local” segment pool makes it easy to export all the segments required by the current setting, if the setting is going to be exported to another system.

Properties of Real Segments

We distinguish between memory segments and PRBS/PRWS segments.

- PRBS/PRWS segments are defined by the polynomial they are calculated from. The width of a PRWS segment (the number of bits per word) is automatically adjusted to the width of the port to which it is assigned (the number of terminals).
- Memory segments consist of vectors and traces. A **vector** specifies all the parallel bits of a port. The serial bit stream of a terminal line is called **trace**.

Data Memory Usage

To understand the data memory consumption of a sequence, it is best to think in data words. A word of an E4832A module consists of 1 to 16 bits, depending on the general segment resolution and the frequency multiplier setting of the specific channel. A word of an E4861A module consists of 16 to 64 bits, while a word of an E4861B module combines 1 to 128 bits. A word of an E4866A or E4867A module consists of 256 bits.

- One word is reserved for internal use.
- A pure PRBS/PRWS segment does not consume memory. Pure PRBS/PRWS segments are directly generated by the built-in shift registers of the modules.

  A special type of a pure PRWS segment is the SFI5 segment. It specifies pseudo random data that is formatted according to the SFI-5 standard. This type of segment can only be used for E4861B modules. These modules are capable of generating not only random data, but also the deskew channel data. For details see *SFI-5 Frame Generator and the SFI5 Data Segment* in the manual *Testing SFI-5 Devices*.

- A distorted PRBS/PRWS segment is produced by the software and downloaded. It is treated like a memory-type segment. It consumes as many words as its polynomial says. A distorted $2^{15}-1$ PRBS, for example, consumes 32767 words. Due to memory constraints, distorted PRBS/PRWS are not available for the polynomials $2^{23}-1$ and $2^{31}-1$.
- Even if you do not use a distorted PRBS, there is a $2^{5}-1$ PRBS allocated internally, which means 31 words are allocated.
- A pseudo segment (Pause0/1, for example) consumes 1 word, if such a segment is used simultaneously at all channels of the module.
- The remaining memory is used for the programmable memory-type data segments.

Segment Type Combinations

The various data generator/analyzer modules have different capabilities.

E4832A Module The E4832A data generator/analyzer module has two sequencers, one for the upper two frontends and one for the lower two frontends. It can simultaneously execute memory type segments on one pair of frontends, and pure PRBS/PRWS segments on the other pair of frontends.

PRBS/PRWS can be combined with Pause0/1, Expected0/1, or Don't Care.

E4861A / B Module The E4861A/B data generator/analyzer module has one sequencer for each of the two frontends. It can execute memory type segments and pure PRBS/PRWS segments in parallel.

Data to Connector Assignment

The algorithm of how the available segment data is assigned to the connectors is as follows:

- The first terminal within a port gets trace 0, the second gets trace 1, and so on.

  The assignment to the connectors depends on what connections you have selected from the terminals to the connectors in the Connection Editor.

- If a terminal is connected to a connector where channels are added, the connector that holds the connection gets the first trace. An added channel gets the next trace.

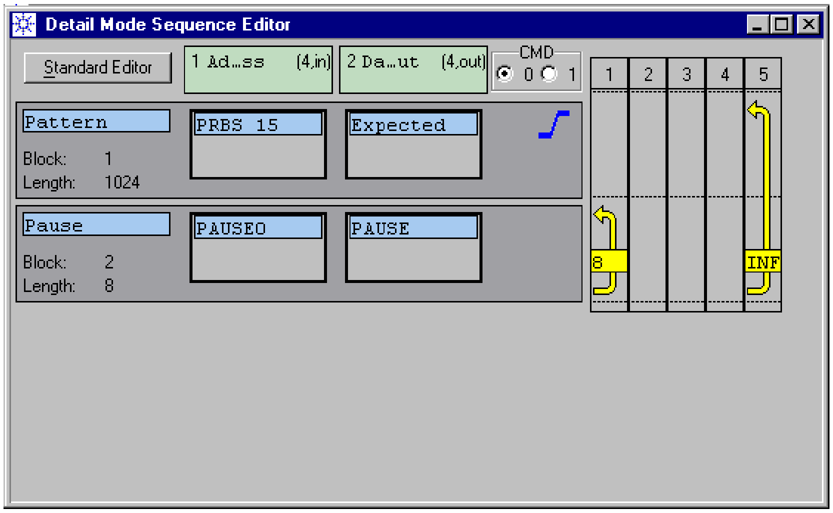
  In this case an exception to the rule “from top to bottom” is made. Added channels are assigned from bottom to top.

Loops

Loops are used to repeat data blocks. For details see *“How to Create and Change Loops” on page 296.*

Looping Example

A sequence looping infinitely 1 Kbit portions of a $2^{15}-1$ PRBS followed by a pause of 64 bits might look as follows:



Pattern
Block: 1
Length: 1024
PRBS 15
Expected

Pause
Block: 2
Length: 8
PAUSE0
PAUSE

Figure 31 **Simple Test Sequence With Loops**

Here, the pause is created by looping eight bits eight times.

If you had chosen a memory-type segment in block one, this segment would start from the beginning with every loop. If the segment is larger than the portion used in the sequence, then there is data that is never generated in this sequence.

However, this is different when looping blocks with PRBS/PRWS segments.

Looping Blocks With PRBS / PRWS Segments

If you do not use the complete PRxS you have chosen for your application but are looping it, then with each new loop cycle the next portion of the PRxS is used.

It can happen that the next portion is the rest of the PRxS and a bit of the beginning. The portioning of the PRxS will go on as long as the looping lasts.

The data representation in the channel memory—in case a distorted 2^{15}–1 PRBS was chosen—and the data stream output of one connector may look as follows:

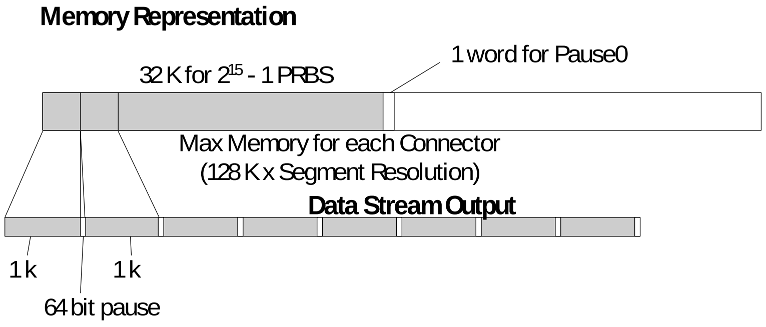

Memory Representation
1 word for Pause0
32 K for 2^{15} - 1 PRBS
Max Memory for each Connector
(128 K x Segment Resolution)
Data Stream Output
1 k
1 k
64 bit pause

Figure 32 PRBS Memory Representation and Generated Data Stream

Hardware Dependencies

The number of blocks and loops that can be used depends on the selected sequencer type (incorporated in the clock module).

Generating Loops With the Clock Module

The E4805B and the E4808A clock module provide:

- 4 counted loop levels (1 performed by the data module, 3 by the clock module).
- Loop counts may be up to 2^{20}. Note that for loop level 1, there is a restriction:

  `LoopCount1 × (blocklength / segment resolution)` $\leq 2^{20}$.

  If this restriction is not met, the loop level 1 can not be used. In this case, the looping has to be performed on a higher loop level.

- 1 infinite loop in level 5.
- Number of blocks + number of loops on level 1 must be ≤ 60.
- Number of blocks + number of loops on levels 2, 3, or 4 must be ≤ 60.
- Minimum block length:
  - segment resolution, if no counted loop is used.
  - 2 × segment resolution, if one loop is starting on level 1, 2, 3, or 4.
  - 4 × segment resolution, if two loops are starting on level 2, 3, or 4.
  - 5 × segment resolution, if loops are starting on level 2, 3, and 4.

Summary of Data-Related Terms

Sequence: The overall stream of generated and expected data, formed by sequence blocks.

Block: A portion of a test sequence which references segments that define generated and expected data. A block refers to all data ports. Its length has to be a multiple of the segment resolution. Single blocks and groups of blocks can be repeated (loops). A trigger pulse can be issued at the beginning of a block.

Segment: Contains the data to be generated or expected: A certain pattern, PRBS, or PRWS. PRxS means algorithmic data. A pattern consists of vectors and traces.

PRBS / PRWS: Pseudo Random Bit/Word data Stream.

Vector: Specifies all the parallel, simultaneous bits of a port within a segment.

Trace: Specifies the serial data transmitted to or expected from a terminal.

Principles of Analyzer Sampling Point Adjustment

The proper comparison of received data with expected data requires that the analyzer captures the incoming data at the right point in time.

That means first of all that the analyzing frontends have to be triggered by a suitable clock frequency.

The sampling frequency may be an issue. Received data usually arrives with the frequency of the stimulating signal or an integer multiple or fraction thereof.

Frequency relations of 2^n

If the analyzer sampling frequency is a 2^n-multiple or fraction of the system clock frequency, this can often be handled within one system. The frequency multiplier provides adequate choices for setting the frequencies of individual ports and channels.

Frequency relations not 2^n

If the sampling frequency is not a 2^n-multiple or fraction of the system frequency (such as 3, 5, 6, 7, 9, and so on), then systems with independent clock modules have to be installed.

They can reside in one and the same mainframe, but if they are to be operated manually, you have to start the user interface individually for each system.

Synchronizing separate systems

Separated generating and analyzing systems and the DUT can be frequency-synchronized to one clock source. The clock source can be the built-in oscillator of the master clock module, an external reference, or even an external source.

For adjusting the sampling start delay and phase, the Agilent 81250 Parallel Bit Error Ratio Tester offers three methods. These methods allow to set the delay before the measurement starts as well as to determine and set the optimum sampling delay automatically.

For details see:

- *"Manual Analyzer Sampling Delay Adjustment" on page 88*—an example of how this is done with the Parameter Editor
- *"Automatic Delay Alignment" on page 90*—how Automatic Delay Alignment works
- *"Automatic Bit Synchronization" on page 92*—the explanation of the principles behind Automatic Bit Synchronization

The automated methods require a special synchronization block within the test sequence. This block should be the first block that specifies expected data.

The synchronization block may be preceded by Pause blocks. Such blocks can be used for establishing a certain delay before the synchronization starts. A delay may be required for giving a PLL time to settle. If the synchronization block is embedded somewhere in a sequence, it should be labeled “START” because the sequence execution begins with the START block.

The synchronization block is automatically repeated until the synchronization criteria are met. The sequencer continues after the analyzers are synchronized.

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial*.

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
  In the *Tutorial*, select *“Analyzer Sampling Point Adjustment”*.

Manual Analyzer Sampling Delay Adjustment

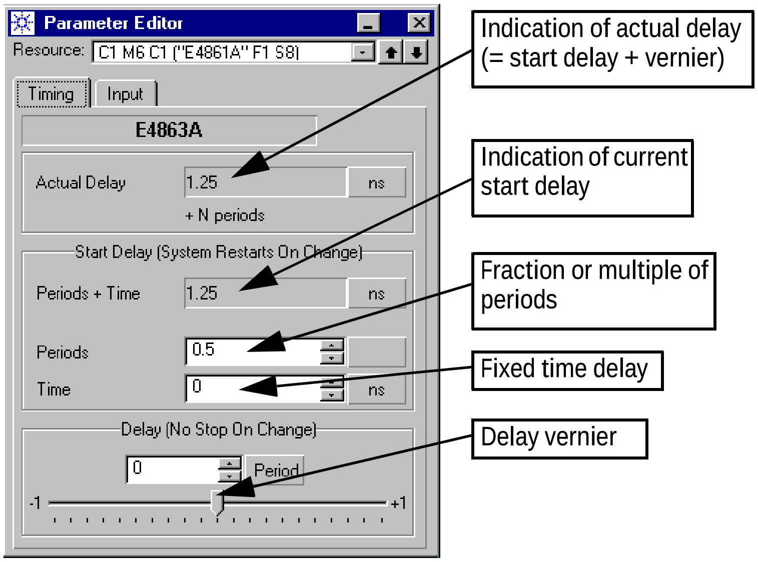
For every analyzer channel, the Parameter Editor allows to specify the start delay between the start of the system clock (usually the start of the generators) and the start of the analyzer. The start delay can be specified as a certain amount of time plus a multiple or fraction of a clock period.

If one of these parameter is changed while a test is running, the test is aborted and automatically restarted.

The analyzer frontends are also supported by a delay vernier. This vernier, implemented as a slider in the Parameter Editor, allows to shift the analyzer sampling delay by up to ±1 clock periods without interrupting a running test.

For E4832A data generator/analyzer modules, the delay vernier requires that the minimum segment resolution is set.

In the user interface, the timing setup looks as follows:



Parameter Editor

Resource: C1 M6 C1 ("E4861A" F1 S8)

Timing | Input

E4863A

Actual Delay 1.25 ns

+ N periods

Start Delay (System Restarts On Change)

Periods + Time 1.25 ns

Periods 0.5

Time 0 ns

Delay (No Stop On Change)

0 Period

-1 +1

Indication of actual delay (= start delay + vernier)

Indication of current start delay

Fraction or multiple of periods

Fixed time delay

Delay vernier

Figure 33 Timing Parameters for a Data Analyzer Frontend

How the Manual Delay Adjustment Works

Manual delay adjustment works as illustrated in the following figure:

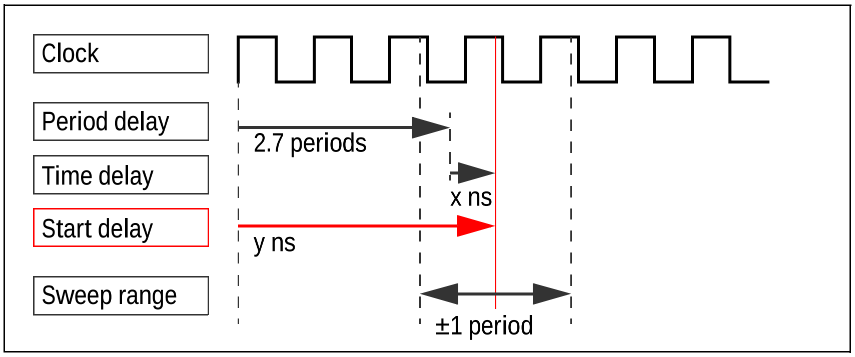


Figure 34 Analyzer Start Delay and Manual Delay Sweep Range

The start delay is composed of period delay and time delay. This delay is used as long as the delay vernier is in zero position.

When the sampling point is moved by means of the delay vernier while a bit error rate (BER) test is running, the BER changes. By observing the actual BER, you can thus measure the eye opening of the received signal.

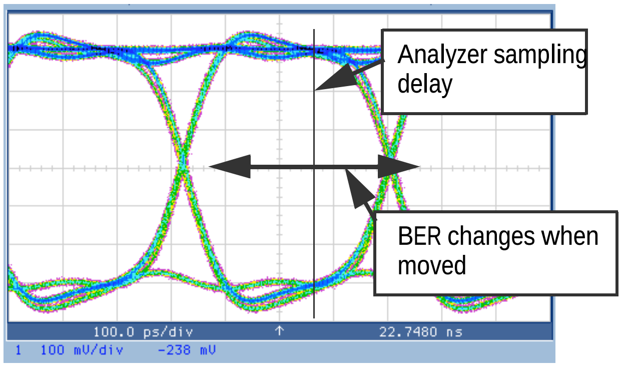


Figure 35 Eye Diagram of a 2.6 GHz Signal

The Parameter Editor always indicates the current delay. Once the width of the eye opening is known, the analyzer sampling delay can be put into optimum position which is in the middle.

Automatic Delay Alignment

Automatic Delay Alignment is used if the expected signal propagation delay can be coarsely specified. The same data must be generated and expected within one sequence block. Memory-based or PRBS data can be used.

How Automatic Delay Alignment Works

The analyzer subtracts a few nanoseconds from the specified start delay and then searches for a sampling point at which bit recognition has an adequate, adjustable accuracy. This accuracy is defined by a bit error rate threshold.

For an E4832A data generator/analyzer module, the search range is limited to ±50 ns from the start delay. For an E4861A module, the search range is limited to ±10 ns.

As soon as the measured BER is below the threshold, the analyzer begins optimizing the sampling delay.

If, during the whole search range, the measured BER is higher than the threshold, the BER counter of this analyzer becomes disabled. This means, a subsequent BER test shows no results for this channel.

The test does not continue unless all analyzers have reported synchronization pass or fail results.

Sampling delay optimization

For optimizing the sampling delay, the analyzer shifts the sampling point in steps and measures the BER at each step. The number and hence the width of these steps is adjustable. It is called **phase accuracy**. The analyzer thus measures the width of the eye diagram and finally positions the sampling point at the optimum, which is in the middle. The duration and precision of this optimization depend on the chosen phase accuracy.

If the delay vernier is not in zero position, its setting is now added to or subtracted from the optimum sampling delay.

The whole process is illustrated in the following figure:

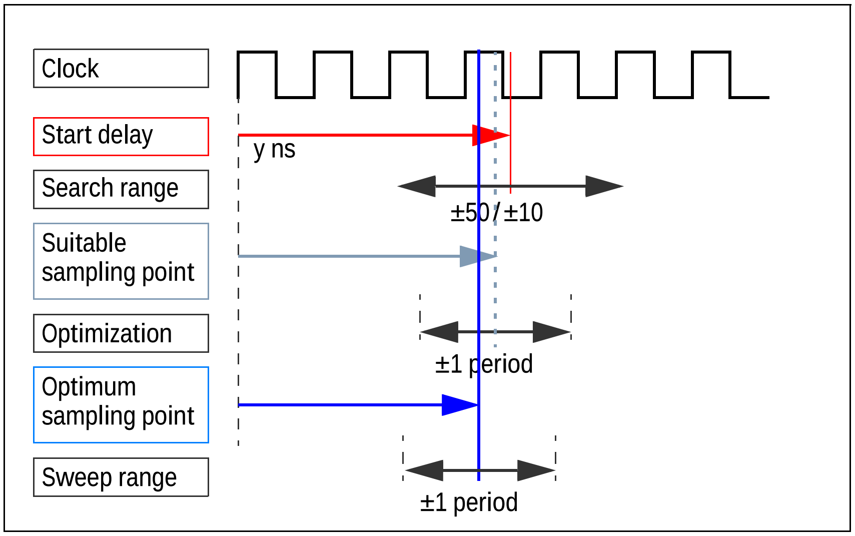


Figure 36 Automatic Analyzer Delay Alignment Process

Auto Delay Alignment result

The resulting absolute delay since starting the test is indicated in the Parameter Editor window of the DUT output port or analyzer frontend (see figure *"Timing Parameters for a Data Analyzer Frontend" on page 89*).

Once the delay has been found, the delay vernier allows to shift the analyzer sampling point by up to ±1 clock periods while the test is running.

NOTE

If you intend to perform measurements after Automatic Analyzer Delay Alignment using the Agilent 81250 Measurement Software, the delay vernier has to be kept in zero position.

Automatic Delay Alignment Flow Control

The flow of Automatic Delay Alignment is illustrated in the figure below:

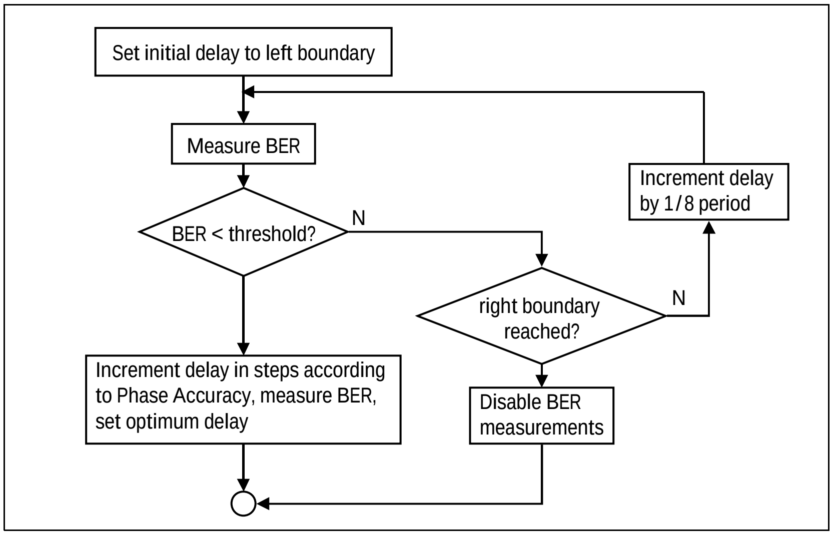

Set initial delay to left boundary
Measure BER
BER < threshold?
N
Increment delay by 1/8 period
right boundary reached?
N
Increment delay in steps according to Phase Accuracy, measure BER, set optimum delay
Disable BER measurements

Figure 37 Automatic Analyzer Delay Alignment Flow Control

Automatic Bit Synchronization

Automatic Bit Synchronization is used to align the incoming data pattern with the expected pattern. This procedure differs from Automatic Delay Alignment as it considers only the phase shift between the analyzer clock and the received data.

Automatic Bit Synchronization offers the option to enable or disable Automatic Phase Alignment:

- Automatic Bit Synchronization without Automatic Phase Alignment is used if the total delay from test start is unknown but a certain edge delay relative to the analyzer clock is expected.
- Automatic Bit Synchronization with Automatic Phase Alignment is used if the delay is completely unknown.

How Automatic Bit Synchronization Works

The analyzer uses the start delay that has been specified with the Parameter Editor to determine the sampling point in relation to its clock. It calculates the sampling point position as “start delay modulo periods”. That means, if the start delay includes a number of full periods, these periods are ignored.

The analyzer then samples the incoming data until the incoming data matches the expected pattern with an adequate, adjustable accuracy. This accuracy is defined by a bit error rate threshold.

In order to minimize the time needed for synchronization, the algorithm takes the kind of expected data—PRBS or memory—into account.

Once the desired accuracy is reached, then the incoming bits are aligned with the expected bits—the analyzer is synchronized with the incoming data.

Auto Bit Sync with Auto Phase Alignment

If Automatic Phase Alignment is enabled, then the analyzer fully automatically adjusts itself to capture the incoming data at the optimum sampling delay.

The analyzer measures the width of the eye diagram and positions the sampling point at the optimum, which is in the middle. Actually, the same optimization procedure as for Automatic Delay Alignment is used.

If the delay vernier is not in zero position, its setting is now added to or subtracted from the optimum sampling point.

The process is illustrated in the following figure:

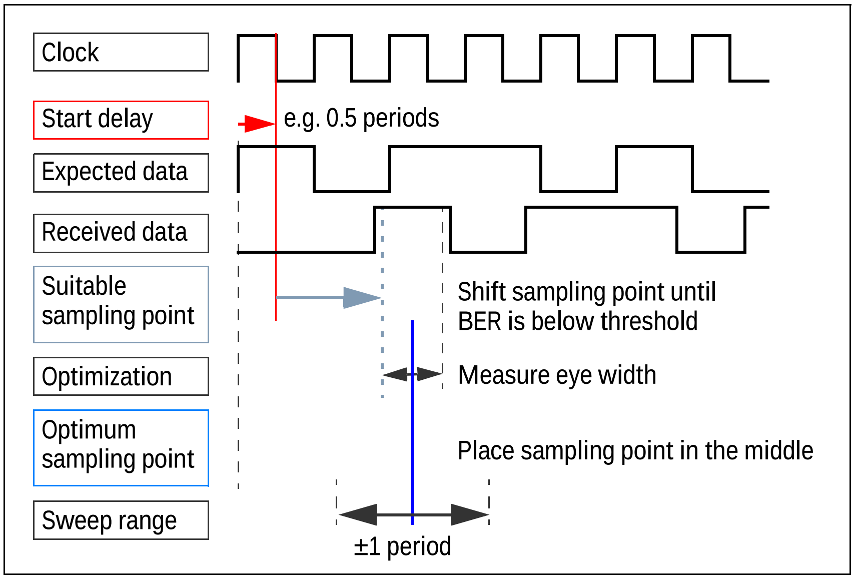


Figure 38 Automatic Bit Synchronization Process

Auto Bit Sync result

The delay found by Automatic Bit Synchronization is indicated by the Parameter Editor (see figure *"Timing Parameters for a Data Analyzer Frontend" on page 89*). This delay is relative to the analyzer's sampling clock. It does not report the absolute delay that would be required between the start of the generator and the analyzer for capturing a complete pattern.

Once the analyzer has been synchronized, the delay vernier allows you to shift the analyzer sampling point by up to ±1 clock periods while the test is running.

NOTE

If you intend to perform measurements after Automatic Bit Synchronization using the Agilent 81250 Measurement Software, you should enable Automatic Phase Alignment. The measurements require that the delay vernier is in zero position.

Synchronization data requirements

Pseudo random data can be sent and expected within one sequence block. A pure analyzing system may also expect memory-type data. Memory data cannot be used for Automatic Bit Synchronization on a single system that generates *and* analyzes data.

The reason is that Automatic Bit Synchronization works differently for PRBS and memory-type data.

The following figure shows the building blocks of an analyzer channel:

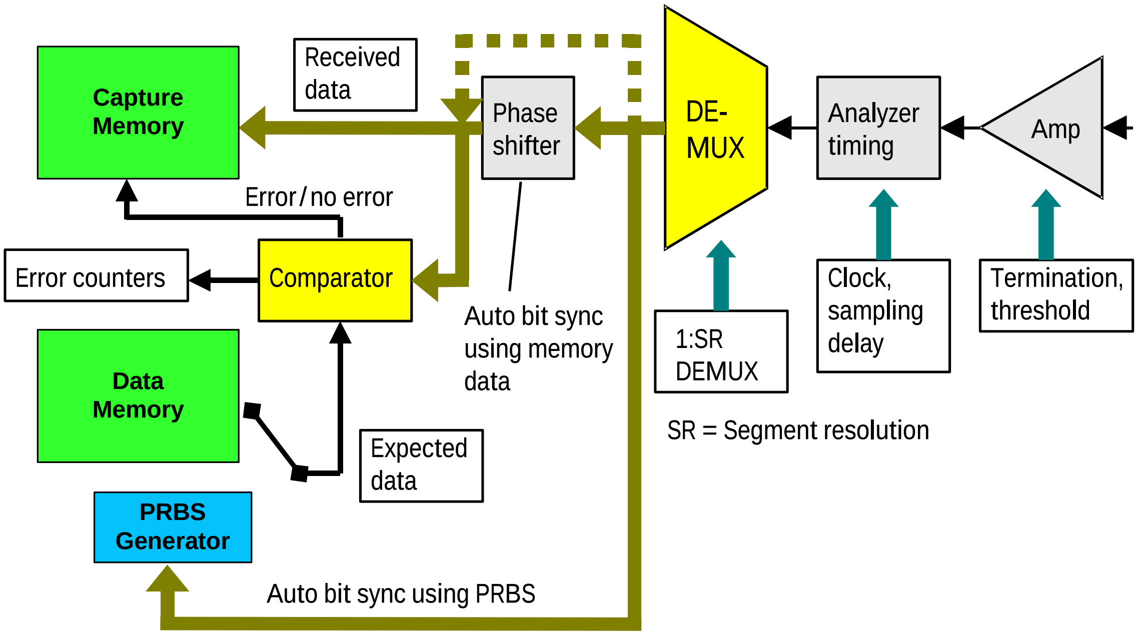


Figure 39 Hardware Architecture of an Analyzer Channel

For synchronizing on pure PRBS data, the built-in PRBS shift register generates the expected data. Note that a distorted PRBS is memory-type data.

For synchronizing on memory-type data, a dedicated phase shifter is used. It is bypassed if pure PRBS data is used for the synchronization.

NOTE Due to the different hardware paths used for the synchronization, it is not possible to synchronize a port on PRBS data and then analyze memory data.

If the test sequence contains for a port a mixture of PRBS and memory data and Automatic Bit Synchronization is used, then the synchronization block of this port must contain memory data. This, in turn, is only possible on a pure analyzing system.

Auto Bit Synchronization Using PRBS Data

The flow of Automatic Bit Synchronization using pure PRBS data is illustrated in the figure below:

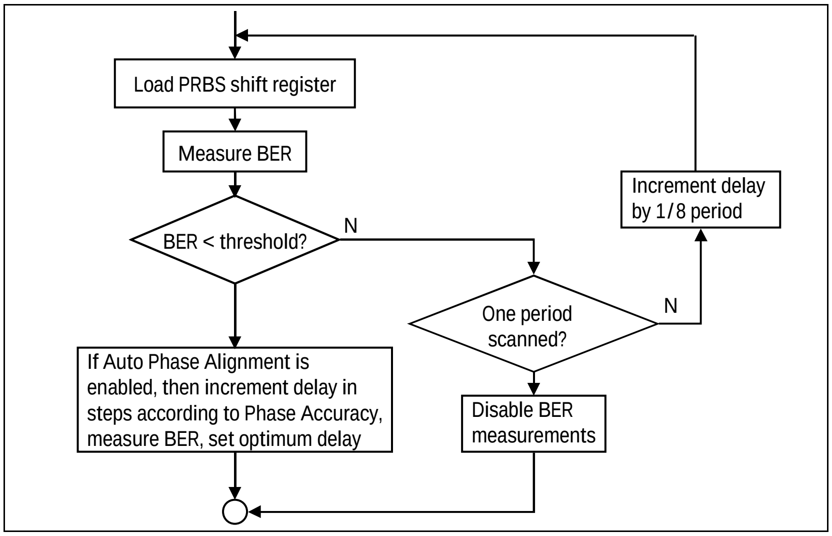

Load PRBS shift register
Measure BER
BER < threshold?
N
Increment delay by 1/8 period
One period scanned?
N
If Auto Phase Alignment is enabled, then increment delay in steps according to Phase Accuracy, measure BER, set optimum delay
Disable BER measurements

Figure 40 **Automatic Bit Synchronization Using PRBS Data**

The PRBS feedback shift register is preloaded by the setting. If the output of the shift register does not match the incoming data, the delay is increased, and the shift register is loaded with n bits of the incoming 2^n-1 PRBS data. After loading the shift register, the expected data should match the incoming data, because identical shift registers are used on the generating and analyzing side.

If during a whole period the measured BER is still higher than the threshold, then the BER counter of this analyzer becomes disabled. This means, a subsequent BER test shows no results for this channel.

Auto Bit Synchronization Using Memory Data

Automatic Bit Synchronization on memory-type data can only be performed on an analyzer system that is independent from the generator system.

If memory-type data is used for Automatic Bit Synchronization, the first 48 expected bits of each analyzer channel have to be unique. That means, this 48-bit word must not appear twice in the segment that is used for synchronization. It must not consist of only zeros or ones, but it may include don’t care bits.

The minimum length of the synchronization block is 32 words which means *32* × *segment resolution* bits.

NOTE A distorted PRBS can be used for Automatic Bit Synchronization. This is memory-type data that requires a block length of 2^n-1 words or *(2^n–1)* × *segment resolution* bits. This length ensures that every repetition of the block contains the same data.

If a data port is being synchronized, then all the detect words of all channels must be found within ±5 times the port’s segment resolution. For example, if the segment resolution is 16, this range is ±80 bits.

Comparison of the Methods

Considering the test duration, manual adjustment of the analyzer sampling delay is the fastest method. When a test is started, all analyzers of the port are already in position.

If the necessary delay is unknown, this can be determined by an initial test using Automatic Delay Alignment.

However, the manual adjustment is only useful for devices with uniform delay at all output terminals.

Each of the two other methods, Automatic Delay Alignment and Automatic Bit Synchronization, has its advantages.

Table 8 Comparison of the Automatic Synchronization Methods

Characteristics	Automatic Delay Alignment	Automatic Bit Synchronization
Sampling point find range:	±50 / ±10 ns, depending on type of module	unlimited
Delay after synchronization:	absolute delay	uncertainty of n periods
Sync data limitations:	none	Memory-based sync data requires: – two systems – first 48 bits unique – min. block length 32 x segment res. Mixed PRBS & memory test patterns require a memory-type sync segment
Synchronization speed:	slower	faster
Working with other PRBS generators:	no	yes

NOTE When testing a demultiplexer with memory-based data, demultiplexer rewiring should be used. For details see *"Automatic Rewiring of Demultiplexer Terminals" on page 119*.

Fast Bit Synchronization

Fast Bit Synchronization is a method for aligning expected data with incoming data very quickly. This method does not change the analyzer sampling delay.

NOTE That means, before Fast Bit Synchronization can be used for a test, the optimum analyzer sampling point has to be determined and set, either manually or by means of one of the automatic methods.

Fast Bit Synchronization has been implemented to support first of all *Recirculating Loop Tests*. These are optical tests where the received pattern is of very limited duration—in the range of 200 µs to 300 µs. For such tests, neither Automatic Delay Alignment nor Automatic Bit Synchronization can be used.

Characteristics

Fast Bit Synchronization can only be used in conjunction with pure PRBS or PRWS data. Such data is directly generated by the module hardware.

Fast Bit Synchronization “aligns” the expected data with the incoming. That means, the built-in PRBS generator of the analyzer(s) is started and its output is compared with the received bits. The synchronization block is not looped. Its minimum length depends on the PRBS polynomial and the individual segment resolution of the port. When it ends, the process has found a start condition, and the real measurement begins.

This requires a sequence setup the Standard Mode Sequence Editor cannot handle. Fast Bit Synchronization requires that the Detail Mode Sequence Editor is used.

Fast Bit Synchronization has no “failed” condition, because the sampling delays and thresholds have to be set beforehand. If these do not fit, the process will find only a limited number of matching bits and will set an arbitrary start point. A following BER test would yield unacceptable error rates.

Restrictions

Pure PRxS data only

During the synchronization phase, the received data must not be all zeros or all ones. The synchronization process might lock onto such patterns. The same PRxS data as expected has to be sent. For the characteristics of PRxS data see also *“Appendix B: PRBS/PRWS Data Segments” on page 425*.

ParBERT 43G systems

ParBERT 43G error detector systems are normally synchronized via *Auto Bit Sync*, combined with *Auto Phase Alignment*. This standard setting also optimizes the sampling delays and thresholds of the 16 analyzers.

If Fast Bit Synchronization is used, you have to know the required analyzer delay and set it manually (see also *“Additional Characteristics of ParBERT 43G Systems” on page 116*). The BER shall be zero.

If the port setting does not suffice, you may also need to fine-tune the sampling delay of each of the 16 analyzers.

Data comparison

In *Compare and Acquire Around Error* or *Compare and Capture* mode, please note: Every received bit is stored, but during the synchronization phase, the comparators are disabled. That means that all bits captured during that time are marked error-free, though errors may have occurred.

Length of the Synchronization Block

The synchronization block can be as long as desired. You can, for example, execute the whole measurement within that block. After the synchronization phase, the comparators are enabled and incoming bits are compared with expected.

However, the length of the synchronization block must not remain under a certain limit. The minimum block length includes the bits required for the synchronization plus enough additional bits to fill at least one additional sequencer clock period.

The minimum block length can be calculated as

```
Block length = SR × (m+2)
```

where

SR = individual segment resolution of the DUT output port (the same as `General Segment Resolution × FM factor of the port`, see also *“Frequency Multiplier and Segment Resolution” on page 68*)

m = data factor (see the table below)

In the following table, a PRxS polynomial $2^n – 1$ is used.

Table 9 Calculation of the Data Factor

Module	Condition	Data factor
E4832A		m = n+1
E4861A	SR ≤ 32	m = n+1
	SR = 64	m = (n + (n modulo 2)) / 2
E4861B	n = 7, 11, 15, 23, 31 and port width is a number in power of two (1, 2, 4, 8, ...)	
	SR ≥ 64	m = 1
	SR = 32	m = 2
	SR < 32	m = n
	Other polynomials, or port width not a power of 2	
	SR ≤ 32	m = n
	SR = 64	m = (n + (n modulo 2)) / 2
	SR = 128	m = (n + (n modulo 2)) / 4
E4867A		m = (n + (n modulo 2)) / 4
E4869A / E4869B	Depends on the data modules (see E4861A, E4861B)	
E4811A	As E4861B	

NOTE The term `(n modulo 2)` is zero if n is even; it is one if n is odd.

Summary of Synchronization-Related Terms

Start delay: Analyzer sampling delay setting with Parameter Editor.

Delay vernier: A slider provided by the Parameter Editor for analyzer frontends plugged into a data generator/analyzer module.

Automatic Delay Alignment: Analyzer sampling point optimization if time window is known. Sets and shows the full delay since start.

Automatic Bit Synchronization: Analyzer sampling phase adjustment based on the BER. Sets and shows the phase delay with respect to the analyzer clock.

Fast Bit Synchronization A method to align expected PRxS data with incoming PRxS data. Does not change the analyzer sampling delay.

Data Capturing and Analysis Principles

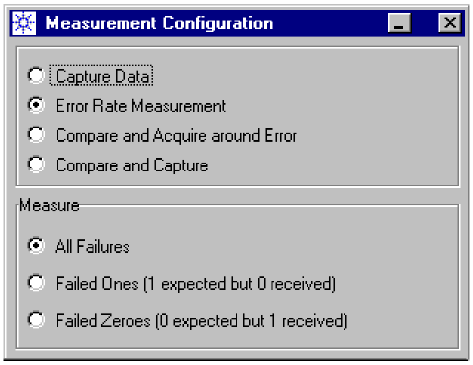
The system provides four test and measurement modes. Tests can be preceded by a synchronization procedure during which the analyzer frontends optimize the position of the sampling delay.

See:

- *"Functional Tests" on page 101*—a summary of the standard tests. Note that the Agilent 81250 ParBERT Measurement Software provides additional test and measurement functions.
- *"Error Analysis and Marginal Tests" on page 103*—reminds you of the various capabilities.
- *"Display of Test Results" on page 103*—an overview of the standard result displays. Note that the Agilent 81250 ParBERT Measurement Software provides enhanced capabilities.

Functional Tests

The functional tests are chosen from the Measurement Configuration window. For details see *"Choosing the Kind of Measurement" on page 261*.



Measurement Configuration

○ Capture Data
● Error Rate Measurement
○ Compare and Acquire around Error
○ Compare and Capture

Measure

● All Failures
○ Failed Ones (1 expected but 0 received)
○ Failed Zeroes (0 expected but 1 received)

Figure 41 Measurement Configuration Window

The functional tests include:

Capture Data

In capture mode, the analyzer frontends capture data until the sequence expires or their memory is filled. The result can be reviewed in a state list and also graphically. Depending on the data generator/analyzer module and the segment resolution, an analyzer can capture up to 2 or up to 8 Mbit of data.

Bit Error Rate Measurement

The Bit Error Rate Measurement scans the received data in real time and shows the resulting actual and accumulated number of bits, the actual and accumulated number of errors, and the actual and accumulated bit error rate. The display is updated approximately every second.

Compare and Acquire Around Error

The Compare and Acquire around Error mode compares and acquires data in real time. The memory capacity in words per channel and the number of bits per word depend on the type of module and the chosen segment resolution or frequency multiplying factor, respectively.

Table 10 Capture Memory

	E4832A	E4861A	E4861B / E4811A	E4866A / E4867A
Storage capacity	128 K words	128 K words	128 K words	128 K words
Bits / word	1 to 16	16 to 64	1 to 128	256

If an error occurs, it is possible to define when the system should stop after the occurrence of the error.

If the system includes E4832A modules, then the minimum value to stop is 976; if E4861A modules are present, the minimum is 3904.

The maximum number is the usable memory, calculated as:

```
Memory (bits) = Storage capacity × Segment resolution
```

The captured data including the errors can be viewed as an Error State list and also graphically with the Waveform Viewer.

It is possible to load expected data segments which have been captured from a reference device or imported from a simulation.

Compare and Capture

The Compare and Capture mode compares and acquires data in real time. It continues until the sequence expires or the Stop button is pressed.

This mode is first of all intended to be used with event handling where the reaction on an error is specified within the sequence.

You can view the result in the Error State Display where errors are highlighted, and also graphically with the Waveform Viewer.

Error Analysis and Marginal Tests

A device can be stimulated with arbitrary input signals using the variable pulse parameters provided by the Agilent 81250 system. These functions are provided by the 675 MHz module and its generator frontends.

Parameters such as levels, delay, and width can be varied independently for each channel or for a DUT port as a whole.

Distorted signals as well as glitches and pulse delay variations can be emulated using the digital channel addition capabilities of the Agilent 81250 system. Up to four channels can be added to emulate a real-time pulse delay variation with up to four phases.

E4838A generator frontends are additionally supported by the analog channel add function which allows to generate signals with overshot and ringing.

For details see *"How to Set Up a DUT Input Port or Generator Channel" on page 226*.

Display of Test Results

Depending on the chosen measurement, several displays are available. For BER tests, use the BER window:

- The Bit Error Rate window shows the current and accumulated results and is continually updated.

Captured data and the results of real-time compare tests can be investigated with the Error State Display and the Waveform Viewer.

- The Error State Display shows the captured data and errors in tabular form. Auxiliary functions are provided that support quick navigation.
- The Waveform Viewer shows the captured data and errors in graphical form. It provides waveform selection as well as markers and zoom for precise waveform analysis.

For details see *"Viewing Generated and Captured Data" on page 349*.

Summary of Analysis-Related Terms

BER: Bit Error Rate. The number of errored bits divided by the number of received bits.

Error State Display: Shows captured data and errors in tabular form.

Waveform Viewer: Shows generated and captured data as well as errors in graphical form. Allows to investigate phase relationships.

Capture Data In capture mode, the analyzer frontends capture data until the sequence expires or their memory is filled.

Compare and Capture The Compare and Capture mode compares and acquires data in real time. It continues until the sequence expires or the Stop button is pressed.

Compare and Acquire Around Error The Compare and Acquire around Error mode compares and acquires data in real time. If an error occurs, it is possible to define when the system should stop after the occurrence of the error.

Event Handling Principles

The Agilent 81250 system can detect a variety of events and react on events.

The reaction may simply be a trigger pulse at the TRIGGER OUTPUT of the clock module, but can also be a change of the test sequence.

See:

- *"Usage of Events" on page 105*—some examples that show the purpose of using events
- *"What is an Event?" on page 105*—explains the various kinds of events
- *"Actions Upon an Event" on page 106*—explains what can be done if an event has occurred

Usage of Events

The system provides several ways to react on events:

- Stop and go:

  This is useful for production tests, where data is sourced to the DUT, a measurement is performed with other equipment, the next data pattern is sourced, and so on.

- Block switching:

  The data sequence is no longer fixed. Based on certain events, certain portions of the overall sequence can be executed.

  This has the advantage that one and the same sequence can be created and downloaded once and then used for several tests. There is no need for re-programming the system.

- Trigger external devices:

  The event can generate a trigger signal at the TRIGGER OUTPUT of the clock module. This can be used to trigger an external instrument like a sampling oscilloscope or logic analyzer to sample the data at an error location.

- Bolt on:

  The Agilent 81250 system can be integrated into a large IC test system. The IC tester would issue a trigger to start the Agilent 81250 system for a special measurement. The Agilent 81250 system would perform the test and return pass/fail information that can be examined and evaluated by the IC tester.

- Match loop:

  PLL-based devices typically require an initialization segment that has to be repeated until the device is synchronized. The event that controls repetition would be “an error occurred”.

For setup examples see *“How Do I Use Events?” on page 392*.

What is an Event?

An Agilent 81250 system equipped with the Agilent E4805B clock module has a great deal of options. It can react on:

- any bit combination of the 8-bit trigger pod (see *“Trigger Pod” on page 55*)
- any bit stream error detected by one of the analyzer frontends
- the status of the VXI ECL trigger lines T0 and T1
- an event triggering command issued locally or remotely

Ten events can be defined—five for immediate actions and five for deferred actions.

Events Causing Immediate Action Actions on such events occur immediately (although there is an internal delay). They can be used to launch a trigger or to abort the test, for example.

Events Causing Deferred Action Actions on such events occur at the end of the current sequence block. If the events come asynchronously, this feature ensures that the current block is properly executed and terminated.

These events are associated with priorities. Event number 5 has the highest priority, event number 1 the lowest.

NOTE Events for immediate action override events for deferred action.

Actions Upon an Event

If an event occurs, the system provides the following options:

- Go to:

  Goes to a certain block in the overall sequence and executes that block. The block is identified by its block label. The implicit “End” block (which is automatically assigned and does not need to be defined) terminates the sequence and hence the test.

- Trigger:

  Launches a trigger pulse to the TRIGGER OUTPUT of the central clock module.

- VXI-T01:

  Sets the VXI ECL trigger lines T0 and T1 to 01, 10, or 11.

All these options can be freely combined.

Summary of Event-Related Terms

Event: A signal that must be responded to.

Event causing deferred action: An event that causes an action at the end of the currently executed sequence block.

Event causing immediate action: An event that is serviced as fast as possible, without waiting for the end of the currently executed sequence block.

Action upon an event: Any combination of the following:

- Go to sequence block.
- Set the TRIGGER OUTPUT of the central clock module.
- Set the VXI ECL T0/T1 trigger lines.

ParBERT 43/45G Systems

The Agilent 81250 ParBERT 43/45G is a solution for generating and analyzing electrical data streams of 38 Gbit/s up to 43.2 Gbit/s and more.

The ParBERT 43G allows you to stimulate and analyze 16:1 multiplexers and 1:16 demultiplexers at data rates of 2.7 Gbit/s and 43.2 Gbit/s, according to the OC-768 and SFI-5 (SERDES Framer Interface 5) data range.

ParBERT 45G uses sixteen 3.35 Gbit/s modules and frontends and can exceed 43.2 Gbit/s.

Data analysis A ParBERT 43G error detector system allows you to determine the bit error rate of transmission lines or serial devices operated at 43.2 Gbit/s. It supports the investigation of FEC devices at 43.01841 Gbit/s including the FEC rate resulting from 255/236 overhead.

Data generation Using a ParBERT 43G pattern generator system, you can send a serial bit stream of up to 43.2 Gbit/s to an OC-768 demultiplexer component.

Using four E4867A 10.8 Gbit/s data analyzer modules, it is then possible to analyze the parallel output of a 1:4 demultiplexer.

Operating principle For data generation, sixteen 2.7 Gbit/s or 3.35 Gbit/s data generators are multiplexed to create a 43/45 Gbit/s data stream. On the analyzing side, a 43/45 Gbit/s data stream is demultiplexed and fed into sixteen data analyzers.

The data streams to be generated or expected as well as the signal frequencies and levels, are controlled by the ParBERT user software.

Details

See:

- *“ParBERT 43/45G Components” on page 108*—a description of the basic 43G bundles
- *“ParBERT 43/45G Configurations” on page 110*—several system configurations, including the controller
- *“ParBERT 43G Software Support” on page 112*—an overview of the special characteristics of ParBERT 43G systems and their representation in the user interface

ParBERT 43 / 45G Components

Four preconfigured bundles are available for the most common 43.2 Gbit/s applications:

- Agilent E4894B 43.2 Gbit/s pattern generator bundle
- Agilent E4896A 45G pattern generator bundle
- Agilent E4895B 43.2 Gbit/s error detector bundle
- Agilent E4897A 45G error detector bundle

All these bundles require an external PC as the system controller. This PC is not included.

High Speed Pattern Generator Bundles

A pattern generator bundle multiplexes sixteen 2.7 or 3.35 Gbit/s data generators to create a 43.2 Gbit/s data stream. The bundles include:

- One 13-slot VXI mainframe
- One IEEE 1394 PC link to VXI
- One E4808A high performance clock module
- *E4894B bundle*: Sixteen 2.7 Gbit/s generator frontends E4862A built into eight E4861A 2.7 Gbit/s data modules

  E4896A bundle: Sixteen 3.35 Gbit/s generator frontends E4862B built into eight E4861B 3.35 Gbit/s data modules
- One 43.2 Gbit/s multiplexer module E4868B, including 32 cables to connect to the generator channels (SMA to MCX), and a cable pair to connect to the DUT (1.85 mm to 1.85 mm)
- 15446A 8-line trigger input pod
- E4875A ParBERT software

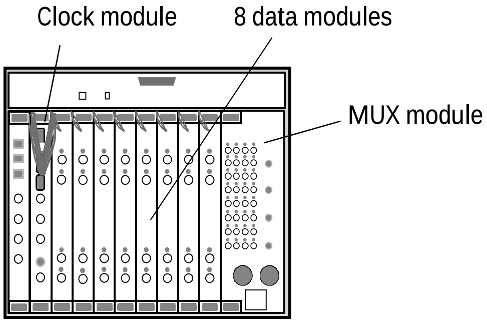


Figure 42 ParBERT 43G Pattern Generator Bundle

High Speed Error Detector Bundles

An error detector bundle demultiplexes a 43.2 Gbit/s data stream and provides the data to sixteen 2.7 or 3.35 Gbit/s data analyzers. The bundles include:

- One 13-slot VXI mainframe
- One IEEE 1394 PC link to VXI
- One E4808A high performance clock module
- *E4865B bundle*: Sixteen 2.7 Gbit/s analyzer frontends E4863A built into eight E4861A 2.7 Gbit/s modules

  E4867A bundle: Sixteen 3.35 Gbit/s analyzer frontends E4863B built into eight E4861B 3.35 Gbit/s data modules
- One 43.2 Gbit/s demultiplexer module E4869B, including 32 cables to connect to the analyzer channels (SMA to MCX), and a cable pair to connect to the DUT (1.85 mm to 1.85 mm)
- E4875A ParBERT software

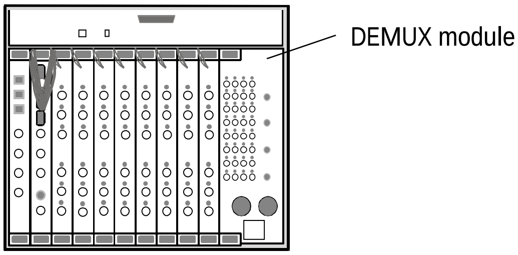


Figure 43 ParBERT 43G Error Detector Bundle

ParBERT 43/45G Configurations

A setup for testing high speed multiplexers *and* demultiplexers would require both bundles. This configuration is illustrated in the figure below.

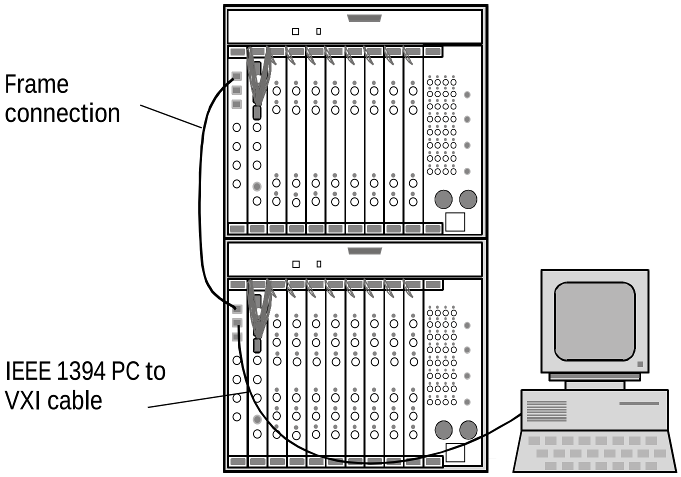


Figure 44 Configuration for Testing Multiplexers and Demultiplexers

A third VXI frame is needed if more than 16 data generators or analyzers are required.

For example, you may need an extra generator for sourcing a clock pulse to the device under test. Or, if you are testing a demultiplexer device with built-in clock recovery circuits, you may need an extra analyzer for conditioning the recovered clock signal.

In such cases it is recommended to add a third frame in the middle between the pattern generators and the analyzers. This frame should house the MUX and DEMUX modules.

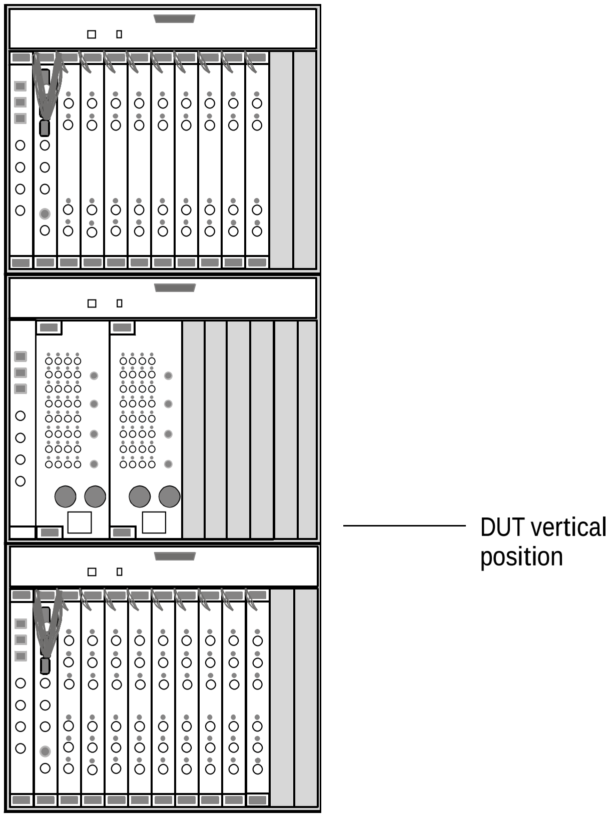


Figure 45 Setup Using More than 16 Data Modules

To keep the test cables as short as possible, the DUT should be placed close to the MUX/DEMUX modules, as illustrated in the figure above.

For the connections between the MUX/DEMUX modules and the generators and analyzers please refer to the *Agilent 81250 ParBERT Installation Guide*.

ParBERT 43G Software Support

The Agilent 81250 software recognizes a ParBERT 43G system automatically. The *Build Systems* process of the Agilent 81250 Configuration Tool identifies the installed E4868B or E4869B module and creates an appropriate system.

For ParBERT 43G systems, many features have been implemented in the software to support quick and easy test setup, and also to prevent the MUX/DEMUX module from damage due to illegal channel parameter settings.

NOTE A MUX or DEMUX module can hardly be compared with a clock module or a data generator/analyzer module, although it combines some features of both basic categories.

You should consider a ParBERT 43G system as a self-contained unit for testing very high speed data multiplexing and demultiplexing components and devices. In this context, setting generator or analyzer channel parameters to individual values is useless and, worse, risks damaging the MUX or DEMUX module.

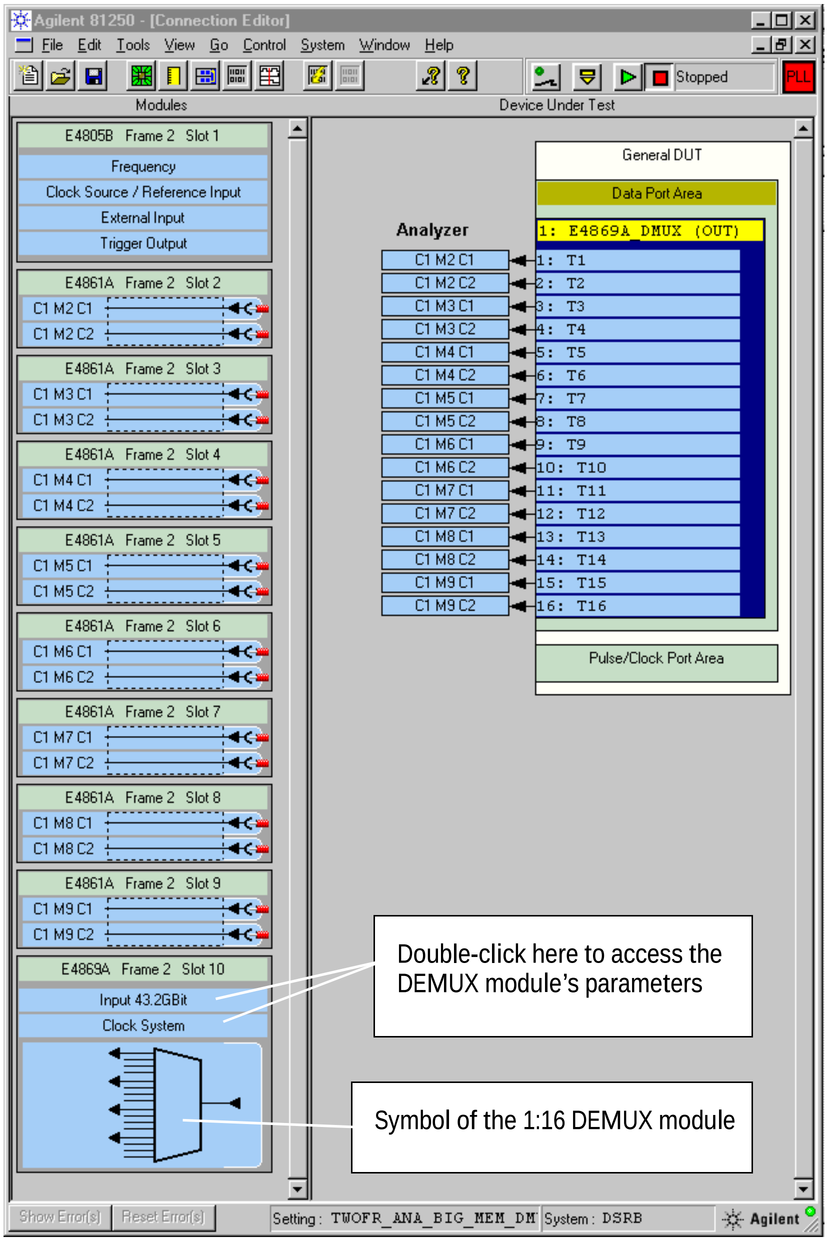
Therefore, most of the usual port and channel parameters are preset to appropriate values and cannot be changed.

For setting up a ParBERT 43G system, two buttons shown in the Connection Editor (see below) provide easy access to the most relevant parameters.

If the system contains additional generator or analyzer channels, these can be configured as usual.

Automatic ParBERT 43G Connections

After starting the user software for a ParBERT 43G system, the Connection Editor window appears as shown in the following figure.



Setting : TWOFR_ANA_BIG_MEM_DM System : DSRB

Double-click here to access the DEMUX module’s parameters

Symbol of the 1:16 DEMUX module

Figure 46 Connection Editor of a ParBERT 43G Error Detector System

The example above refers to a ParBERT 43G error detector system. The software grabs the first 16 analyzer frontends following the master clock module and connects them with the DEMUX module.

NOTE This is the way the ParBERT 43G hardware has to be connected, even if certain modules contain analyzer *and* generator frontends!

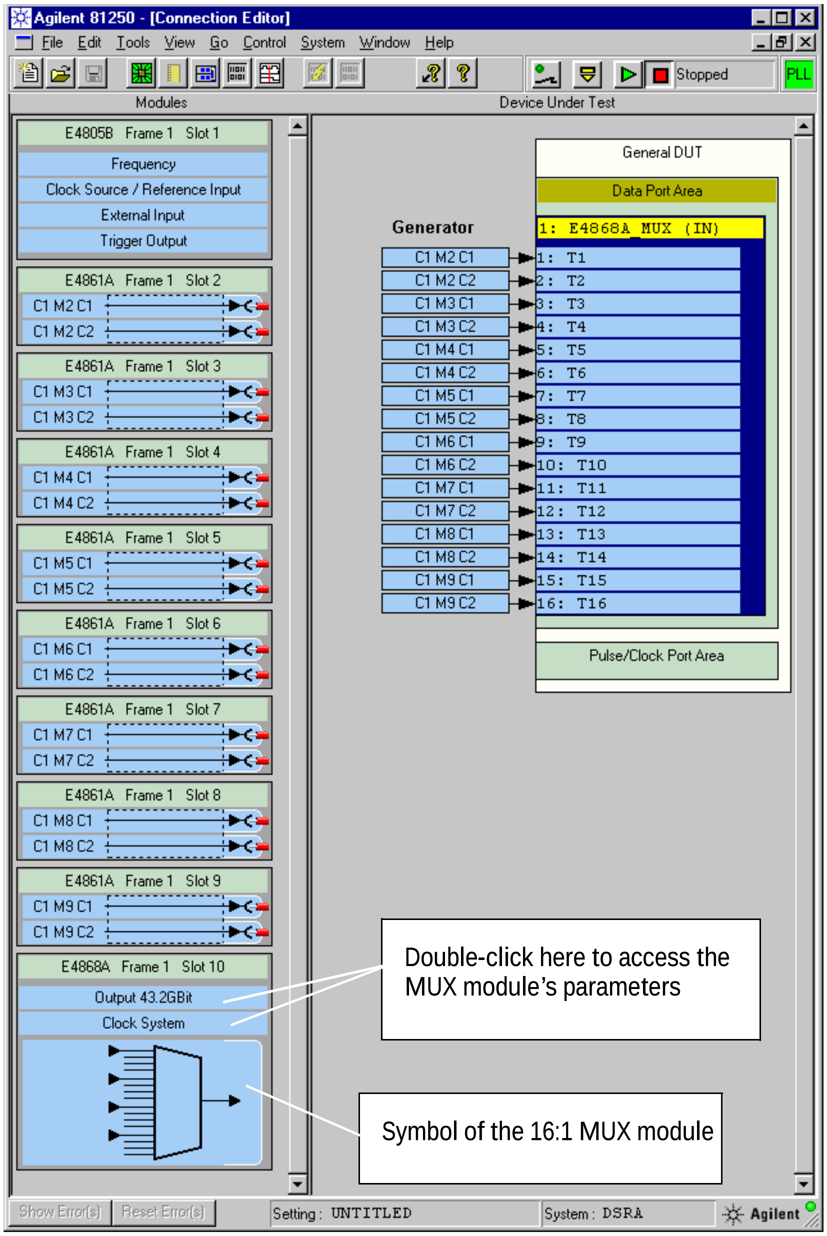
The DUT section shows the parallel side of the DEMUX module, because this provides the signals to the analyzers. Sixteen differential output port terminals are readily connected to and will be analyzed by 16 differential analyzer frontends. Two buttons allow to access the properties of the DEMUX module.

This is a limited perspective, of course. The real DUT is the device that is connected to the DEMUX module and generates a high speed serial data stream.

The real DUT may also generate a recovered clock which has to be conditioned and used as an external source clock for the analyzing system. If this is the case, the analyzing system needs an additional analyzer. This, in turn, requires an additional data module and hence an expander frame.

But if the hardware is present, additional ports and terminals can be set up with the Connection Editor.

The Connection Editor window of a ParBERT 43G pattern generator system looks rather similar.



Double-click here to access the MUX module’s parameters

Symbol of the 16:1 MUX module

Figure 47 **Connection Editor of a ParBERT 43G Pattern Generator System**

Here, 16 differential generator frontends are connected to 16 differential MUX module input terminals.

ParBERT 43G <=> ParBERT Conversion

The Connection Editor does not allow you to disconnect the frontends from the MUX/DEMUX module.

If you wish to use a ParBERT 43G pattern generator or error detector system for other purposes, you have to disconnect and de-install the MUX/DEMUX module completely and use the “Build Systems” function of the Agilent 81250 Configuration Tool (see the *Agilent 81250 Installation Guide* or *“How to Set the Operating Mode” on page 135*).

After that, you have a normal, general-purpose ParBERT system with 16 generator or analyzer channels that can be used for stimulating a multiplexer or measuring the output of a demultiplexer device.

This approach has been taken to prevent the MUX/DEMUX module from damage and to protect your investment.

CAUTION

Never attempt to operate an E4868A MUX or E4869A DEMUX module in a general-purpose ParBERT system! These modules are highly susceptible to overvoltage and electrostatic discharge.

If you have re-installed a MUX/DEMUX module, make sure that you perform the *Build Systems* operation of the Agilent 81250 Configuration Tool **before** starting the Agilent 81250 user software.

Build Systems reverts the system to a ParBERT 43G with all its built-in protection mechanisms.

ParBERT 43G Port and Channel Parameters

All parameters of the 16 channels connected to a MUX/DEMUX module are preset to suitable values, so that a quick test setup and start is ensured. Only a limited number of parameters can be changed.

For the parameters of the MUX/DEMUX module (basically frequency and amplitude) see *“How to Set Up a 43G MUX/DEMUX Module” on page 210*.

Additional Characteristics of ParBERT 43G Systems

Some auxiliary functions are also locked or preconfigured at ParBERT 43G systems:

No global delay offset

- A global *Delay Offset* for the master clock module is not supported.

No generator start delay

- On a ParBERT 43G pattern generator system, an individual start delay setting (data port timing) is not supported.

Auto Bit Sync and DEMUX Rewiring

- On a ParBERT 43G error detector system, automatic *Analyzer Synchronization* is always enabled, including the modes *Auto Bit Sync*, *Auto Phase Alignment*, and *DEMUX Rewiring*.

  You can disable the automatic *Analyzer Synchronization*. This, however, should only be done, if you have chosen the measurement mode *Capture Data*. If *Capture Data* is selected, you cannot specify expected data. But automatic analyzer delay adjustment would require expected data.

NOTE

If you have disabled the automatic *Analyzer Synchronization*, you have to know the required analyzer delay and set it manually. To set the delay, you can use the delay vernier on the Timing page of the port.

If you don’t know the delay, you should first run a test with expected data and automatic *Analyzer Synchronization*, and then check the timing on the Timing pages of the analyzer channels.

The results should enable you to specify a global port delay.

- If *Analyzer Synchronization* is enabled, Auto Bit Synchronization is performed (see also *“Automatic Bit Synchronization” on page 92*), independent of the kind of expected data. Memory-type data must provide a suitable detect word for each channel.

  Automatic Delay Alignment is not supported, because the signal delay caused by the E4869A DEMUX module is a priori unknown.

- DEMUX Rewiring (see also *“Automatic Rewiring of Demultiplexer Terminals” on page 119*) is enabled, but only performed if memory data is expected.

  If the DEMUX Rewiring parameters shall be checked or changed, this can be done by opening the Detail Mode Sequence Editor and selecting the *Sync* item from the context menu.

Zero adjust and cable deskew

- In the Deskew Editor, the functions for *Zero Adjust* and *Cable Delay* measurements of generators or analyzers connected to a MUX or DEMUX module are disabled.

  But deskewing is required in the following cases:

  - After repair or exchange of a module or frontend
  - You wish to use a ParBERT 43G pattern generator system for stimulating a multiplexer device

- You wish to use a ParBERT 43G error detector system for analyzing the response of a demultiplexer device

If deskewing is required, this can be done after removing the MUX/DEMUX module and performing the *Build Systems* operation of the Agilent 81250 Configuration Tool.

NOTE Before removing the MUX/DEMUX module, you have to disconnect it from the data generator/analyzer modules. It is strongly recommended to unscrew the cables from the generator/analyzer modules and to leave the connectors at the MUX/DEMUX module in place, because the cables have been calibrated. If you disconnect both sides, you should label the cables so that they can be re-installed for the same connections.

For details, please refer to the *Agilent 81250 Installation Guide*.

After removing the MUX/DEMUX module and performing the *Build Systems* operation, you can align the connectors, attach your test cables, and compensate for cable and propagation delays (see *"How to Compensate for Internal and External Delays" on page 364*).

Clock data recovery (CDR)

- The DEMUX module of a ParBERT error detector system contains also a clock data recovery circuit (CDR).

  The CDR allows to recover the demultiplexer clock from the incoming data stream. The CDR includes an own phase locked loop. If the CDR could not synchronize on the incoming data, the green PLL indicator on the user interface turns red and shows the letters "CDR" (see also *"PLL Lock Indicator" on page 152*).

Support of Multiple User Interfaces

Using both Agilent 81250 ParBERT 43G bundles means using two ParBERT user interfaces. This is required for independent clock generation and parameter setup.

The Agilent 81250 Configuration Tool and the ParBERT user software greatly support the use of a number of user interfaces:

- More than one user interface can be automatically started.
- Every user interface can be individually configured.

  The configuration parameters for each user interface include: Location of the firmware server (local or LAN address), name of the system to be operated, name of the setting to be automatically downloaded to the system.

- Every user interface can easily be switched to operate one of the configured systems.
- Tests can be started and stopped simultaneously on two or more user interfaces and hence systems.

  For this purpose, the software includes two utilities (see *"How to Use the System Starter Utilities" on page 385*).

Automatic Rewiring of Demultiplexer Terminals

Demultiplexer rewiring is a special feature for testing demultiplexers. When you are testing a demultiplexer, you apply serial data to one terminal and analyze parallel data from a number of terminals.

If you are sourcing pure PRBS data, then the demultiplexer generates a PRBS of the same polynomial at each of its terminals. These signals may have a time offset, but the system is able to handle this and synchronize the analyzers correctly.

NOTE DEMUX rewiring should be used when testing a demultiplexer with memory-based data. DEMUX rewiring is required for demultiplexers with "unpredictable behavior". That means, you do not know at which pin the first bit arrives.

The functions for automatic analyzer delay adjustment (Automatic Delay Alignment or Automatic Bit Synchronization) will not work and any test will fail if such demultiplexers are tested without DEMUX rewiring.

DEMUX rewiring has the following impact on a test:

- Rewiring increases the system's response time on test start.
- Rewiring changes the order of terminals in the output ports.
- All output ports are rewired (with the same scheme—therefore all output ports have to have a similar setup).

DEMUX Rewiring Overview

Many demultiplexers have the peculiarity that one cannot predict at which of the output terminals the first bit of the serial input bit stream appears. Moreover, stopping the test and starting it again with the same serial bit stream might have the effect that another terminal becomes the bit-number-one output.

NOTE This is also a characteristic of the E4869A 43.2 Gbit/s DEMUX module built into ParBERT 43G error detector systems. Therefore DEMUX rewiring has to be used when testing high-speed devices with memory data on a ParBERT 43G error detector system.

This behavior does not affect the order of the bits at the outputs. The order is the same as in the input bit stream.

The Problem

A demultiplexer with a non-predictable behavior causes a problem when you specify the data segments for the output port (for the analyzers of the test system). Every column (trace) of the segment refers to a terminal of the output port. Trace 1 of the segment is expected from the first terminal, trace 2 from the next, and so on.

If the terminal at which the bit number one appears is not known, it is also unknown which trace of the segment has to hold the expected data.

An example is illustrated in the figure below:

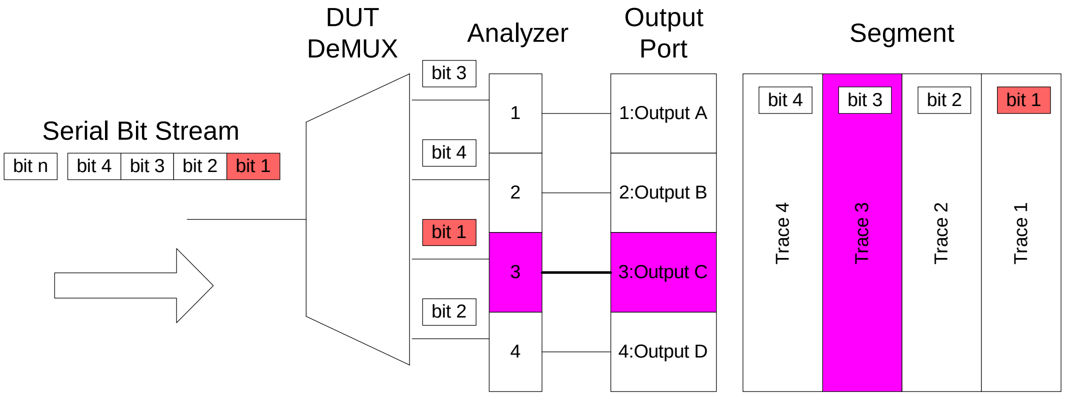


Figure 48 Result of Unpredictable Demultiplexer Behavior

Bit 1 of the serial bit stream was assumed to arrive in trace #1 of the segment. But the demultiplexer randomly assigned bit 1 to its third output terminal. The third analyzer receives the bit and assigns it to terminal #3 which is named “Output C”. Terminal #3 in turn is associated with trace #3 of the segment. So the received bit does not match the expected.

The Solution

The solution is DEMUX rewiring. The expected data is specified as usual, assigning the bit number one to the first output terminal.

The software analyzes the current state of the demultiplexer and reacts in a way that establishes the connection between the constant definition of the expected data and the random output assignment of the demultiplexer.

Each time a test is started the system examines the current data flow. This examination is done in iterations until the current state of the demultiplexer is fully determined.

As a result, the order of the terminals is changed. This changes the assignment of the analyzer frontend connectors to the traces of the data segment. It ensures that the first terminal—the one numbered “1”—receives the bit number one and can thus be compared with trace 1 of the segment.

The principle is illustrated in the figure below:

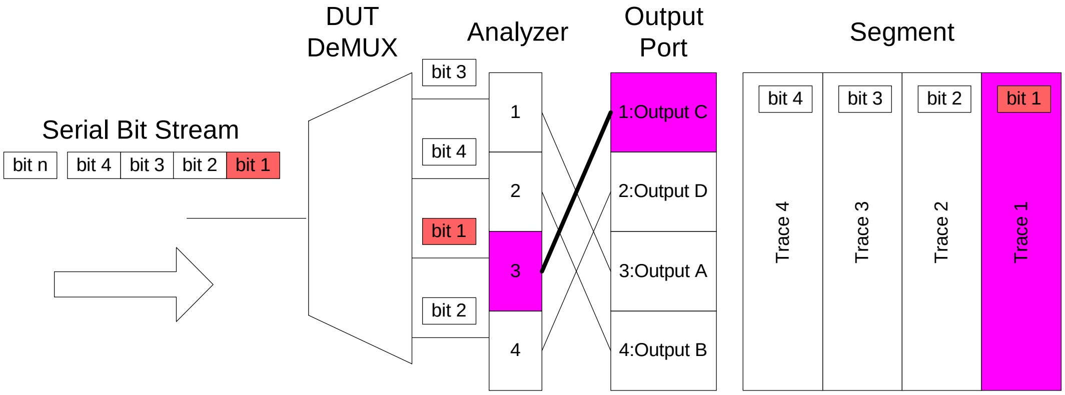


Figure 49 Principle of DEMUX Rewiring

Only the order of the terminals has changed. The connections between the connectors of the analyzers and the terminals of the port are still the same.

Bit 1 of the serial bit stream still arrives at the “Output C” of the demultiplexer (and hence at the same connector of the analyzer). But this time it arrives at terminal #1. This terminal addresses trace #1 of the segment. For this test, all the bits received from “Output C” will be compared with trace #1 and all the bits from “Output D” with trace #2. Then follow the remaining terminals.

Now the data received from the first terminal is compared to the data stored in the first trace of the segment.

DEMUX Rewiring Modes

Demultiplexer rewiring requires that Automatic Delay Alignment or Automatic Bit Synchronization is activated (see also *“Principles of Analyzer Sampling Point Adjustment” on page 87*). It needs and uses the synchronization block at the beginning of the test sequence.

Two algorithms have been implemented for detecting the correct order of the terminals:

- Terminal roundtrip
- Trace detection

Terminal Roundtrip

Terminal roundtrip checks one terminal sequence after the other in order to find coincidence between the expected and the received data.

For each repetition, the order of the terminals within the port is permutated. The possible permutations depend on the demultiplexer’s architecture. For a simple (one stage) demultiplexer the order is just cyclically rotated. This is why this method is called *terminal roundtrip*. In case of a multi-stage demultiplexer architecture, all the possible permutations are tried out.

The process is repeated until all terminals synchronize properly.

Characteristics

The idea of this method is quite straightforward. There is nearly no optimization. This is the reason why this method takes its time, especially for a complicated demultiplexer architecture. New data is downloaded for every repetition. Multi-stage demultiplexers may require a large number of permutations.

If the demultiplexer has an illegal output assignment, this is detected in every case.

Trace Detection

Trace detection uses detect words, formed by the first 48 bits of every trace contained in the “DUT-out” segment of the synchronization block.

Prerequisites

These first 48 bits of every trace have to be unique within the whole segment. Every trace has its own detect word. This is the same requirement that applies to Automatic Bit Synchronization with memory-based data. It is illustrated in the figure below.

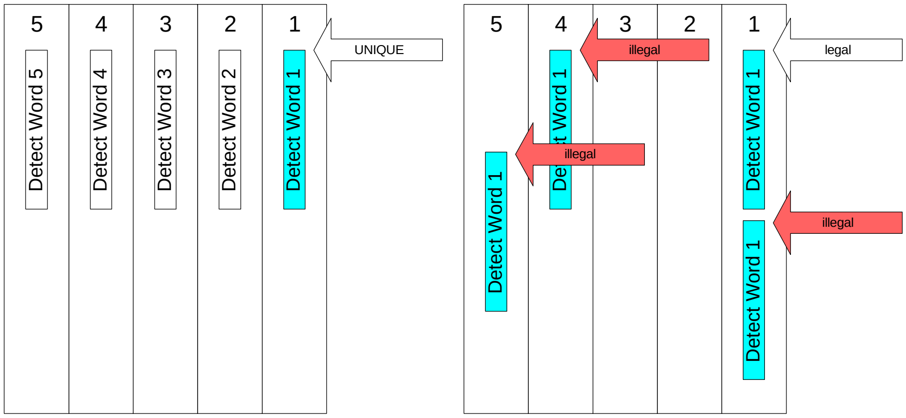


Figure 50 **Unique and Unequivocal Detect Words**

Note that every trace needs an own detect word. This means, the detect word must not appear twice, neither in its own trace nor in any other trace at any position.

Operating principle

The current detect word is attached to every output terminal. A measurement is started. That terminal which has proven to be successful is the output that carries the current detection trace. Depending on the demultiplexer architecture, further detection traces are selected and tested. Finally, the order of the output terminals is rearranged according to the insights made during the trace detect phase. Then, the user’s test sequence is restored and started.

Characteristics

Compared to the terminal roundtrip method, this is a strong optimization. The expected number of rewiring cycles is significantly lower. And the time for rewiring is much shorter.

Once the demultiplexer architecture is known, only the necessary traces are used for detection.

Also, if the demultiplexer assigns the detection traces to wrong outputs, the resulting order of rewired terminals can get into a state a correct demultiplexer never would work with. In such cases, terminal roundtrip is more effective.

Demultiplexer Architecture

A simple demultiplexer has just one stage—one rotating switch that assigns every incoming bit successively to the output terminals in fixed order.

Multi-Stage Demultiplexers

The DEMUX rewiring feature is also able to rewire multi-stage demultiplexers. An example is shown in the figure below:

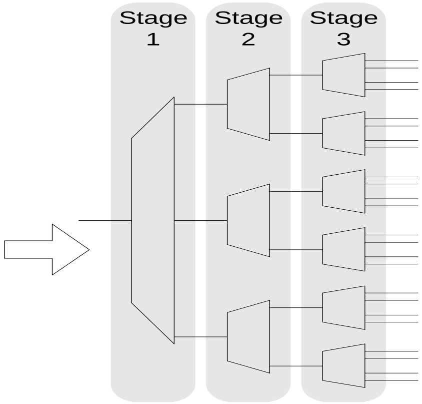
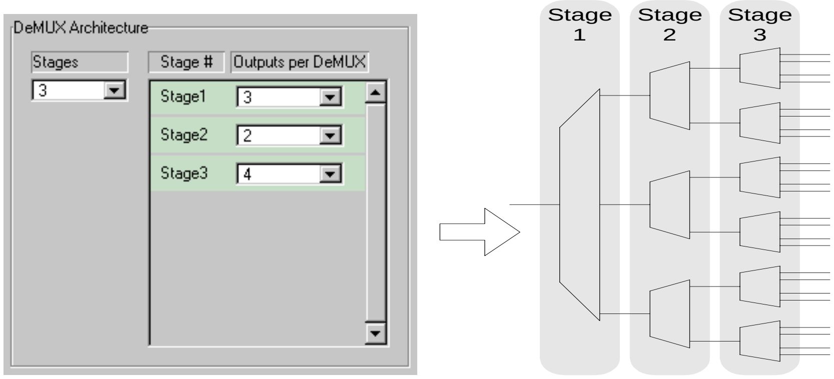


Figure 51 Three-Stage Demultiplexer

On stage one, the number of outputs per demultiplexer is three. On stage two, the number of outputs per demultiplexer is two. On stage three, the number of outputs is four.

This is the way to enter the demultiplexer architecture in the *DeMUX Architecture* panel of the Analyzer Synchronization dialog–first the number of stages, then for each stage the number of outputs per demultiplexer.



DeMUX Architecture

Stages: 3

Stage #	Outputs per DeMUX
Stage1	3
Stage2	2
Stage3	4

Figure 52 DeMUX Architecture Entry Panel

As you can see, you need to know the structure of the demultiplexer. The number of *Required Terminals* (which is 24 in the example above) is automatically calculated.

For details see *"How to Synchronize an Analyzer With Incoming Data" on page 272.*

Only symmetrical architectures are supported. This means that all demultiplexers of one stage have to be identical.

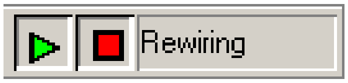
NOTE The number of *Stages* as well as the number of *Outputs per DeMUX* have an impact on the number of possible permutations and hence on the duration of the rewiring and synchronization process. The number of possible permutations is displayed as *Worst Case Rewiring Cycles*.

The expected duration that is also displayed is only a rough estimation, assuming a synchronization segment of moderate length. The actual duration may differ.

For the *trace detection* algorithm, the number of rewiring cycles for a multi-stage demultiplexer is calculated as *number of terminals* + *1*.

For the *terminal roundtrip* algorithm, all possible permutations are taken into account. Every demultiplexer of every stage may exhibit that unpredictable behavior. Depending on the demultiplexer architecture, terminal roundtrip may take half a minute or even hours.

TIP As the user interface is blocked during that time, it is recommended to check the number of *Worst Case Rewiring Cycles* before starting the test.



The rewiring phase is indicated by the side of the Run/Stop buttons.

Error messages are displayed if required parameters or terminals and/or analyzers are missing. *Overflow* is displayed if the estimated time exceeds 24 hours.

Rewiring a ParBERT 43G Error Detector System

A ParBERT 43G error detector system (see *“ParBERT 43/45G Systems” on page 107*) analyzes the output of an E4869A DEMUX module.

This 1:16 demultiplexer has the unpredictable output assignment behavior described above. Therefore, DEMUX rewiring has to be activated when using the E4869A module together with memory-based data.

The recommended rewiring options are:

- *Synchronization*: ON
- *Auto Bit Sync*: enabled
- *DeMUX rewiring*: ON
- *Rewiring Mode*: Trace Detection
- *DeMUX Architecture*: Number of *Stages*: 1
- *DeMUX Architecture*: *Outputs per DeMUX* on *Stage 1*: 16

Test Development Overview

The development of a device test is an iterative process:

1. Set up the test.
2. Run the test.
3. Check the test results.
4. Modify test parameters.
5. Repeat steps 2 to 4 until the results are adequate.
6. Save the final setting for reuse.

All these steps are supported and simplified by the graphical user interface of the Agilent 81250 Parallel Bit Error Ratio Tester.

This chapter provides an overview:

- *"Procedure for Setting Up the Test" on page 128*
- *"Procedure for Running the Test" on page 130*
- *"Procedure for Viewing Test Results" on page 131*
- *"Procedure for Saving the Test Setting" on page 132*

NOTE Once you have created a suitable test setting and verified that the bit error rate is below an acceptable level, you can also execute the measurements provided by the Agilent 81250 ParBERT Measurement Software. These measurements can be performed even if the Agilent 81250 User Software is not active.

Procedure for Setting Up the Test

To set up the test for a new device, it is recommended to perform the following steps in the given order:

1 Study the Device Under Test (DUT).

Identify its input, output, clock, and trigger or strobe terminals. Gather information about its electrical, logical, and frequency characteristics.

In fact, this is the most important step of all.

2 Start the Agilent 81250 system. It comes up with the Connection Editor and the default setting which is called “untitled”.

If you had already set up the instrument and DUT, you would now load the appropriate setting.

For details see *“Open Setting” on page 154.*

3 To create a new setting, construct an image of the DUT on the screen and connect the DUT pins to the connectors of the generator and analyzer frontends.

This is supported by the Connection Editor, which by default shows an image of the instrument and an empty template for modeling DUT input and output pins.

For details see *“Connecting the DUT” on page 195.*

4 Set the global system parameters.

These parameters refer to the central clock module and cover items like clock source, clock frequency, use of an external trigger, control of the built-in trigger generator, etc. Data ports as well as pulse port terminals and unconnected channels can be set to fractions or multiples of the system clock rate.

The tool for setting all kinds of parameters is the Parameter Editor. For details see *“Setting Global System Parameters” on page 175.*

5 Set the characteristics of the input and output connectors.

The characteristics include parameters like voltages, delays, impedances, binary data representation, and so on. This is also done with the Parameter Editor, which in turn can be run conveniently from the Connection Editor.

For details see *“Setting Up Ports and Channels” on page 223.*

6. Decide what kind of test you wish to perform.

   Open the Measurement Configuration window. Choices are bit error rate measurement, capture and compare, or just capture DUT output data.

   For details see *“Choosing the Kind of Measurement” on page 261*.

7. Create the stream of generated and expected data.

   For this purpose, the software provides three Sequence Editors: the Standard Mode Sequence Editor, the Detail Mode Sequence Editor, and the Data/Sequence Editor. These editors enable you to create and maintain the data blocks that form the test sequence.

   The Standard Mode Sequence Editor supports easy setup of bit error rate measurements. The other two editors allow to create an arbitrary sequence.

   For details see *“Creating the Stream of Generated and Expected Data” on page 265*.

8. Create the data segments referenced by the blocks.

   Each block of a sequence contains data segments that specify the generated and expected data. Stored segments can be chosen from lists. New segments can be created with the Segment Editor.

   For details see *“Creating and Editing Segments” on page 309*.

9. Connect the DUT physically to the instrument.

   Use the Connection Editor to ensure that all physical connections match the image on the screen.

10. Setting up the test of a new device sometimes requires that you change cables, add modules or frontends, or change the DUT board. In this case you should compensate the setup for different signal propagation and cable delays. This can be done with the Deskew Editor. See *“How to Compensate for Internal and External Delays” on page 364*.

Now you are ready to run the test.

Procedure for Running the Test

After you have finished the setup:

1. Download the test sequence to the modules.

   This is done by clicking the Prepare button. The download procedure checks whether the test sequence is formally correct and can be executed.

   As downloading a complex sequence can take some time, this is also recommended before running a test that is to be started by a trigger.

   For details see *"How to Download the Test Sequence" on page 346*.

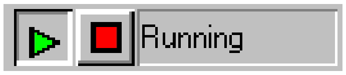
2. If the test has been set up for measuring the bit error rate, open the Bit Error Rate Measurement Display window.

   For details see *"How to View BER Test Results" on page 347*.

3. Click the Run button.

   If the test is set up to be controlled by an external start trigger, it will now wait for this trigger. If not, it starts immediately.

   The test will run until the test sequence is executed or the capture memory is full or, if it is controlled by an external stop trigger, until the trigger is set—whichever comes first.



4. If the test sequence includes an infinite loop, stop the test by clicking the Stop button.

Procedure for Viewing Test Results

If you are running a bit error measurement, the Bit Error Rate Measurement Display window shows you actual and accumulated results. The display is updated every second.

For details see *"How to View BER Test Results" on page 347*.

If you have been running one of the other tests:

1 Open the Error State Display.

  The Error State Display shows the captured DUT output data. If you have been running one of the compare tests, it shows also the deviations between the captured and expected data. Various address and data formats support the investigation.

  For details see *"How to View Captured Test Results" on page 350*.

2 Open the Waveform Viewer.

  The Waveform Viewer enables you to display generated and captured data as well as compare results graphically in a variety of formats.

  For details see *"How to View Waveforms" on page 356*.

Procedure for Saving the Test Setting

It is recommended to save the test setting repeatedly during test development. This ensures that whatever occurs you can always return to the last saved test configuration.

As the system provides different options for different kinds of measurements, it is recommended to save every measurement configuration, such as bit error rate or compare and capture, in its own file.

To save a new setting:

1. Open the *File* menu.
2. Choose *Save Setting As*.
3. Enter a filename that gives some information about the purpose of the setting.
4. Confirm.

To save the current setting occasionally:

♦ Click the Save Setting button.

NOTE Note that settings can also be exported to another directory or drive and imported from there (see also *“Export/Import of a Setting” on page 378*).

System Start and User Interface

The Agilent 81250 user interface basically consists of a window frame, several editors for test setup, and several windows for displaying the test results.

The menu bar on top of the window frame can be used to access the individual windows and to operate the system. Shortcuts are provided by the tool bar buttons.

This chapter provides basic information on both the windows and the options of the main menu. See:

- *"How to Start the System" on page 134*—the explanation of the Agilent 81250 Configuration Tool and the Agilent 81250 User Interface Configuration
- *"Overview of the Windows" on page 145*—a summary of the tools that are combined in the ParBERT user interface
- *"Operating the User Interface" on page 147*—some general tips and keyboard shortcuts for operating the system
- *"Items of the Main Window" on page 150*—the description of the Agilent 81250 main window, its indicators, icons, and menus.

How to Start the System

At a factory-configured Agilent 81250 system with built-in controller, the Windows NT automatic log-in script is enabled. After power on, you are automatically registered as user DVT and the Windows desktop appears.

How to Start the Agilent 81250 Software

The Agilent 81250 software can be run in one of three modes:

- Local
- Controlled
- Remote

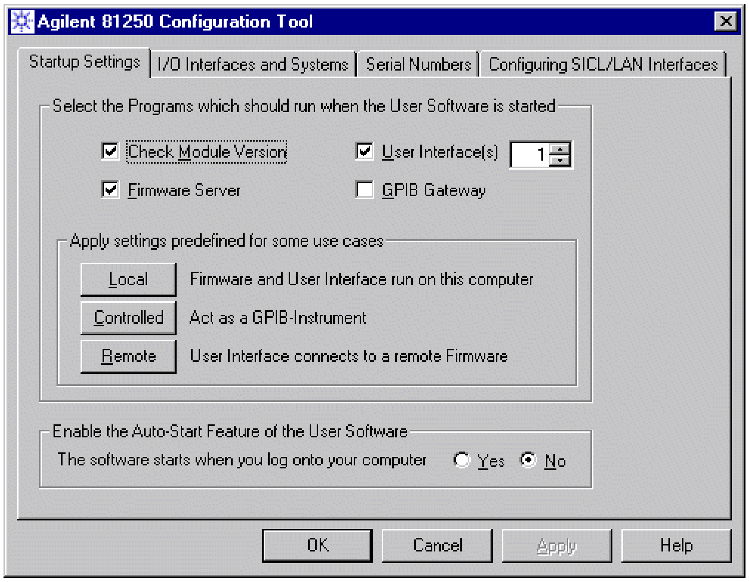
How to Set the Operating Mode

To change the system’s operating mode:

1 Double-click the Agilent 81250 Configuration icon.

Figure 53 Agilent 81250 Configuration Icon

This opens the Agilent 81250 Configuration Tool:


Agilent 81250 Configuration Tool
Startup Settings
I/O Interfaces and Systems
Serial Numbers
Configuring SICL/LAN Interfaces
OK
Cancel
Apply
Help

Select the Programs which should run when the User Software is started

☑ Check Module Version ☑ User Interface(s) 1

☑ Firmware Server □ GPIB Gateway

Apply settings predefined for some use cases

Local — Firmware and User Interface run on this computer

Controlled — Act as a GPIB-Instrument

Remote — User Interface connects to a remote Firmware

Enable the Auto-Start Feature of the User Software

The software starts when you log onto your computer ○ Yes ● No

Figure 54 Agilent 81250 Configuration Window

Startup Settings

2 Select the operating mode you wish to use from now on. Click one of the predefined mode buttons.

Choices are: *Local*, *Controlled*, or *Remote*.

– *Local*

This mode starts both the Agilent 81250 firmware server and the user interface. It is used if the hardware shall be controlled by and the user interface shall be run on the same computer.

– *Controlled*

This mode starts the Agilent 81250 firmware server only. It is used if the hardware shall be controlled by the built-in or external controller, but the user interface shall run on a remote computer. The system can then be operated via GPIB or LAN.

If GPIB shall be used, the GPIB Gateway must be enabled. See also *“How to Control the GPIB Gateway” on page 143*.

- *Remote*

  This mode starts the Agilent 81250 user interface only. It is used on a remote computer to operate an Agilent 81250 firmware server which is in controlled mode.

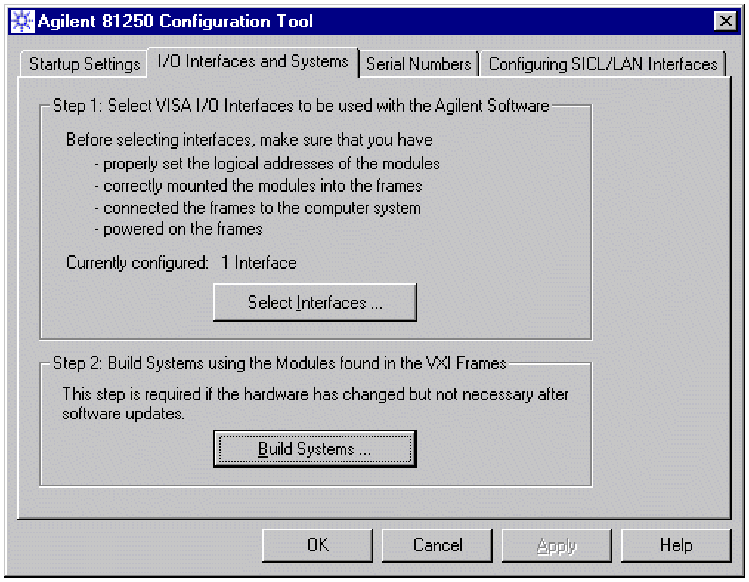
When you are clicking the mode buttons, the checkboxes indicate which software components will be started. You can also specify your own customized mode of operation by clicking the checkboxes.

3 Choose from the following options:

- *Number of User Interfaces*: You can specify that more than one user interface is opened as soon as the user software is started. This applies to manual as well as automatic start.
- *GPIB Gateway*: If you intend to start the system in controlled mode and operate it via LAN, you can disable the *GPIB Gateway*.
- *Start Automatically* starts the Agilent 81250 user software fully automatically. The setting takes effect as soon as the DVT user logs in.

I / O Interfaces and Systems

4 The second page of the Agilent 81250 Configuration window allows you to reconfigure the tester if the hardware has been changed.


Agilent 81250 Configuration Tool
Startup Settings
I/O Interfaces and Systems
Serial Numbers
Configuring SICL/LAN Interfaces
Select Interfaces ...
Build Systems ...
OK
Cancel
Apply
Help

Step 1: Select VISA I/O Interfaces to be used with the Agilent Software

Before selecting interfaces, make sure that you have
- properly set the logical addresses of the modules
- correctly mounted the modules into the frames
- connected the frames to the computer system
- powered on the frames

Currently configured: 1 Interface

Step 2: Build Systems using the Modules found in the VXI Frames

This step is required if the hardware has changed but not necessary after software updates.

Figure 55 Configuring I / O Interfaces and Systems

- The default *VISA I/O interface* is VXI0. If you are using an external controller connected to more than one IEEE 1394 PC to VXI module, you have to select the interfaces you wish to control.
- *Build Systems* checks the available modules and creates new configuration files *dvtsys.txt* and *dvtits.txt*. The present files are

saved as *dvtsys.bak* and *dvtits.bak*. If the mainframe contains more than one master clock module, additional systems (DSRB, DSRC, ...) are automatically set up.

NOTE The Configuration Tool also creates for each real system an offline system and additionally some demo systems.

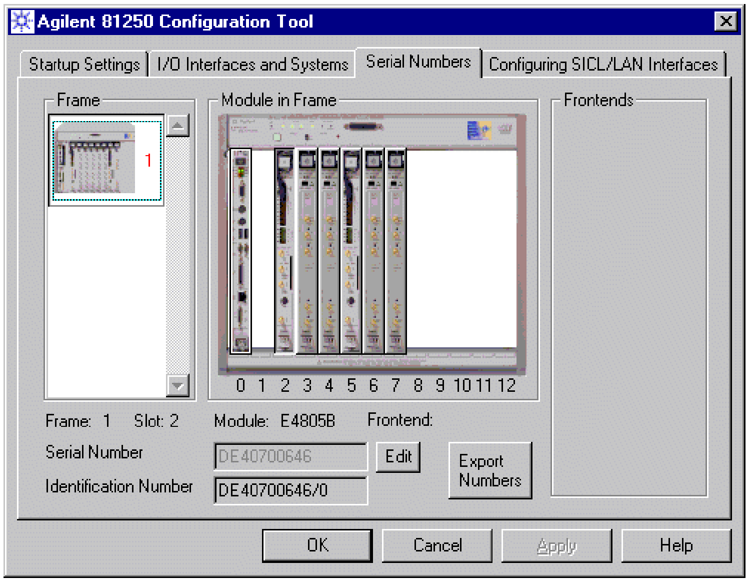
An offline system reflects the actual system configuration. As it does not access the hardware, it can be operated on any PC and used for training purposes.

The demo systems provide simple system configurations and can be used for demonstrating ParBERT features to newcomers.

For more details please refer to the *Agilent 81250 Installation Guide*.

Serial Numbers

The Serial Numbers page of the Configuration Tool can be used to query and archive the serial numbers of the installed modules and frontends.



Frame: 1 Slot: 2

Module: E4805B

Frontend:

Serial Number DE40700646

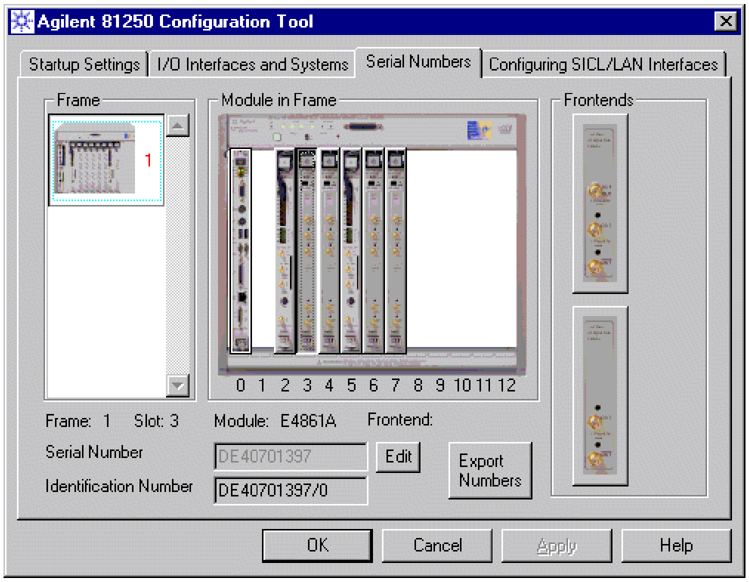
Identification Number DE40700646/0

Figure 56 Serial and ID Numbers of the Master Clock Module

By default, the numbers of the master clock module are displayed.

The *Identification Number* is the hardware manufacturing number. The *Serial Number* can be edited. This may support your asset administration. If you have received a module from repair which has a new number, you can assign the old serial number to that module.

To view the type and numbers of a different module, click on that module.



Frame: 1 Slot: 3
Module: E4861A
Frontend:
Serial Number: DE40701397
Identification Number: DE40701397/0

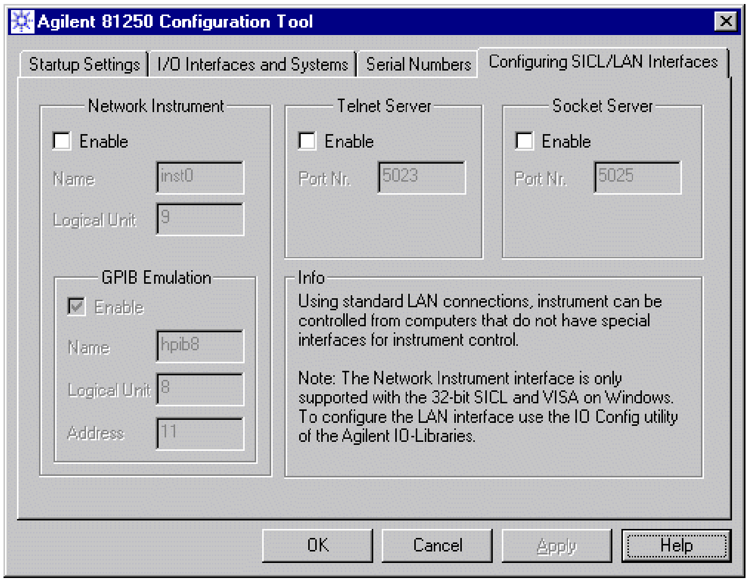
Figure 57 Serial and ID Numbers of a Data Generator / Analyzer Module

To view the type and numbers of a frontend plugged into a data module, click the frontend.

To archive the present system configuration in an ASCII file, click *Export Numbers*.

Configuring SICL / LAN Interfaces

This page of the Configuration Editor allows you to specify LAN connections for remote programming.


Agilent 81250 Configuration Tool
Startup Settings
I/O Interfaces and Systems
Serial Numbers
Configuring SICL/LAN Interfaces
OK
Cancel
Apply
Help

Network Instrument

□ Enable

Name: inst0

Logical Unit: 9

GPIB Emulation

☑ Enable

Name: hpib8

Logical Unit: 8

Address: 11

Telnet Server

□ Enable

Port Nr.: 5023

Socket Server

□ Enable

Port Nr.: 5025

Info

Using standard LAN connections, instrument can be controlled from computers that do not have special interfaces for instrument control.

Note: The Network Instrument interface is only supported with the 32-bit SICL and VISA on Windows. To configure the LAN interface use the IO Config utility of the Agilent IO-Libraries.

Figure 58 Configuration of SICL / LAN Interfaces

You can enable and specify Network, Telnet, and Socket connections. For details please refer to the *Agilent 81250 ParBERT LAN Programming Guide*.

How to Start the System in the Chosen Operating Mode

To start the system:

1 Double-click the Agilent 81250 User Software icon.

Figure 59 Agilent 81250 Start Icon

When you start the user software for the first time, the User Interface Configuration dialog appears.

2 Configure the user interface.

TIP Once you have set the characteristics of the user interface, you can disable the User Interface Configuration dialog. Otherwise, it appears at every software start.

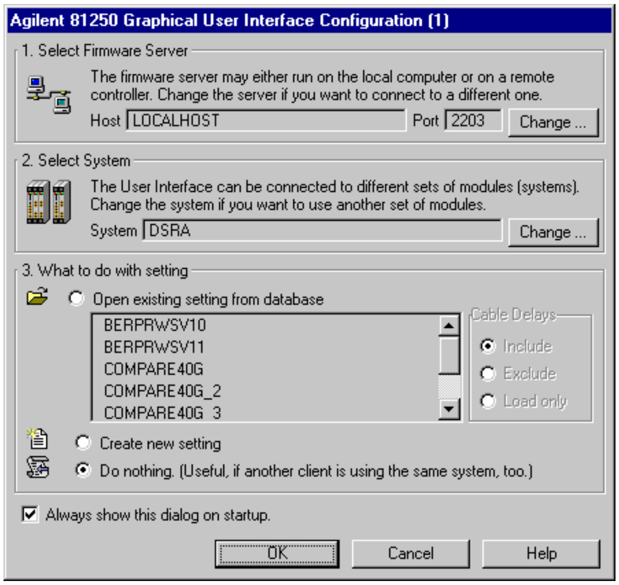
How to Configure the User Interface

You can start the user software more than once—either manually or automatically after logon. In this case you get several user interfaces, and each of them has to be configured.

For example, it is not possible to operate one and the same system from two user interfaces.

By using two user interfaces, you can operate two **independent** systems through one firmware server running on one ParBERT controller. You can also communicate with different firmware servers of different systems which are connected to the LAN.

The configuration window looks as shown below:



Agilent 81250 Graphical User Interface Configuration (1)

1. Select Firmware Server

The firmware server may either run on the local computer or on a remote controller. Change the server if you want to connect to a different one.

Host LOCALHOST Port 2203 Change ...

2. Select System

The User Interface can be connected to different sets of modules (systems). Change the system if you want to use another set of modules.

System DSRA Change ...

3. What to do with setting

○ Open existing setting from database

BERPRWSV10
BERPRWSV11
COMPARE40G
COMPARE40G_2
COMPARE40G_3

Cable Delays
● Include
○ Exclude
○ Load only

○ Create new setting

● Do nothing. (Useful, if another client is using the same system, too.)

☑ Always show this dialog on startup.

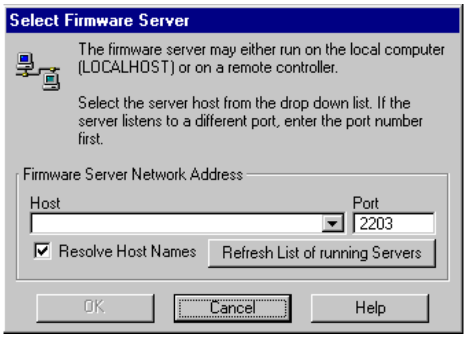
Figure 60 Agilent 81250 User Interface Configuration Window

How to Select the Firmware Server

If you are running the ParBERT user software in *local* mode, the firmware server is found on the same PC which is called LOCALHOST.

If you are running the ParBERT user software in *remote* mode, LOCALHOST is not available. You have to specify the network node on which the firmware server is running. This requires that the ParBERT controller has been connected to the LAN.

1 In the User Interface Configuration dialog, click the *Change* button.



Select Firmware Server

The firmware server may either run on the local computer (LOCALHOST) or on a remote controller.

Select the server host from the drop down list. If the server listens to a different port, enter the port number first.

Firmware Server Network Address

Host:

Port: 2203

☑ Resolve Host Names

Figure 61 Selecting a PC on which the Firmware Server is Running

2 Choose the host from the drop-down list, or enter the computer name or the IP address of the ParBERT controller you wish to connect to.

3 Enter the port number to which the firmware server of that controller listens.

You can update the list of hosts by clicking *Refresh List of running Servers*. Depending on the checkbox, the list shows names or IP addresses. Note that only hosts running in the same subnet as the computer running the ParBERT user interface are recognized.

How to Select the System to be Operated

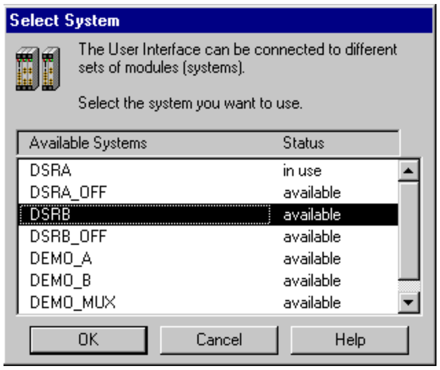
The basic (default) system is DSRA.

You may wish to operate a different ParBERT system from this user interface.

If you start the user interface more than once, you have to choose a different ParBERT system for each user interface.

To select the system:

1 In the User Interface Configuration dialog, click the *Change* button.



Select System

The User Interface can be connected to different sets of modules (systems).

Select the system you want to use.

Available Systems	Status
DSRA	in use
DSRA_OFF	available
DSRB	available
DSRB_OFF	available
DEMO_A	available
DEMO_B	available
DEMO_MUX	available

Figure 62 **Selecting a System**

2 Choose from the list of available systems.

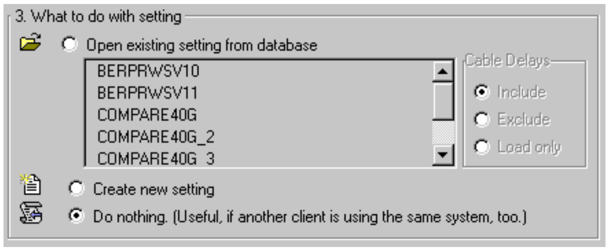
The list includes also demo and offline systems (see *“How to Set the Operating Mode” on page 135*).

Systems already in use by an active user interface cannot be selected.

How to Specify a Start Setting

This section of the Agilent 81250 User Interface Configuration dialog allows to load one of the stored settings of the chosen system automatically together with the user interface.

That means, you need only start the user software and all systems of the Agilent 81250 Parallel Bit Error Ratio Tester are ready for testing a particular device.



3. What to do with setting

○ Open existing setting from database

BERPRWSV10
BERPRWSV11
COMPARE40G
COMPARE40G_2
COMPARE40G_3

Cable Delays

● Include
○ Exclude
○ Load only

○ Create new setting

● Do nothing. (Useful, if another client is using the same system, too.)

Figure 63 **Selecting a Setting**

To load one of the stored settings automatically:

1 Enable the checkbox.

2 Choose from the list.

3 Decide on loading also the cable delays.

How to Disable the User Interface Start Dialog

The user interface configurations are automatically stored. They are identified by numbers, starting from one.

Once the Agilent 81250 Parallel Bit Error Ratio Tester has been set up, you may wish to disable the User Interface Configuration dialog. Otherwise, it appears at every software start.

1. To suppress the dialog, disable the *Always show this dialog on startup* checkbox.

   When the user interface is active, the User Interface Configuration dialog can always be started from the *File* menu, item *Configuration*.

   Even if the User Interface Configuration dialog has been disabled, you can still force the software to show it on startup.

2. To display the dialog as the first window, hold the Shift key depressed when the message “Starting User Interface ...” is displayed.

How to Control the GPIB Gateway

If the GPIB gateway has not been disabled, it is automatically activated when the system is configured to be controlled by another computer and then started.

When the GPIB gateway is active, the GPIB to Agilent 81250 Gateway control panel can be displayed from the Windows task bar.

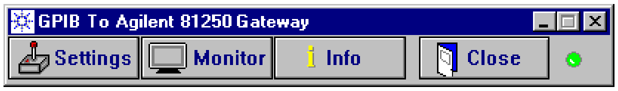
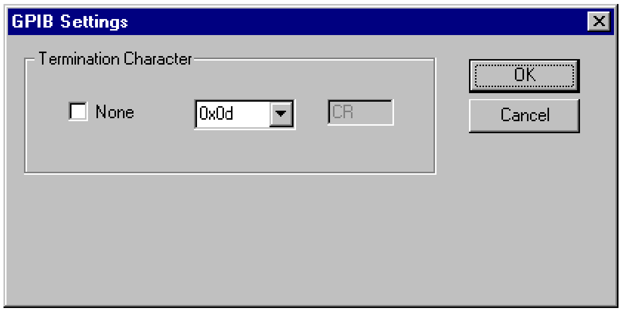


Figure 64 GPIB to Agilent 81250 Gateway Control Panel

The options are:

- *Settings*: Used to specify the termination character for received commands. Transmitted commands and responses are automatically terminated with LF ($0A_{Hex}$), according to IEEE 488.2.

Choices are *none* or any ASCII character between 0 and 127 (decimal). The current termination character is displayed in hex format and alphabetical notation.



GPIB Settings

Termination Character

□ None

0x0d

CR

Figure 65 **GPIB Settings**

The desired character can be entered in decimal, hexadecimal, or octal format. Examples:

Table 11 **Termination Characters**

Character	Hex	Decimal	Octal
HT	0x09	9	011
LF	0x0a	10	012
FF	0x0c	12	014
CR	0x0d	13	015

These usual characters can also be chosen from the pull-down menu.

- *Monitor*: Used to monitor the command transfer in case of problems.

  Shows the commands passing through the receive and send buffers.

- *Info:* Displays details of the session and GPIB setting.

- *Close*: Terminates the GPIB to Agilent 81250 gateway.

  You will be asked whether you wish to terminate the Agilent 81250 server as well.

Overview of the Windows

There are windows for setting up a test and windows for displaying the test results.

See:

- Overview of Test Setup Windows

  A summary of the various editors that can be used for setting up a test

- Overview of Test Result Windows

  A summary of the alternatives the user interface provides for displaying test results

These are described on the following pages.

Overview of Test Setup Windows

The test setup windows are:

- Connection Editor—for reproducing (modeling) the properties of and physical connections to the device under test.
- Parameter Editor—for setting test parameters, such as frequencies, delays, voltages, etc.
- Measurement Configuration—for specifying the kind of test, such as bit error rate or capture and compare.
- Standard Mode Sequence Editor—for specifying one infinitely looped block of data containing the segments of generated and expected data for every data port.
- Detail Mode Sequence Editor—for specifying an individual sequence of blocks which contain the generated and expected data segments and loops. Also used for specifying events and reactions on events or triggers to be generated.
- Segment Editor—for creating the data segments to be generated or expected within the data blocks.
- Data/Sequence Editor—a combination of Sequence and Segment Editor.
- Channel Configuration Editor—for adding generator channels to produce a combined signal.

- Event Edit and Branch windows—for specifying events and actions upon events.
- Deskew Editor—for synchronizing the module connectors and compensating for signal propagation delays caused by cables or the DUT board.

NOTE Most of these windows can be opened by clicking a button of the tool bar, except:

- Channel Configuration Editor—a subfunction of the Connection Editor.
- The Sequence Editors—Standard Mode Sequence Editor, Detail Mode Sequence Editor, Data/Sequence Editor—all accessible from the *Go* menu.
- Parameter Editor and Deskew Editor—accessible from the *Go* menu.
- Event Edit and Branch windows—a subfunction of the Detail Mode Sequence Editor.

Overview of Test Result Windows

A set of windows is provided to present the test results in different views. To access these windows, open *Result Displays* in the *View* menu and select the desired item.

The available test result windows depend on the chosen type of measurement:

- Bit Error Rate—only available, if “Error Rate Measurement” has been selected. Shows the error rate and is updated every second.
- Compared Data—only available, if “Compare and Capture” or “Compare and Acquire around Error” has been selected. Shows captured data and highlights deviations from expected data.
- Captured Data—not available, if “Error Rate Measurement” has been selected. Shows captured data.
- Errored Data—only available, if “Compare and Capture” or “Compare and Acquire around Error” has been selected. Shows zeros (no error) and ones (errors).
- Waveform—not available, if “Error Rate Measurement” has been selected. Shows the captured waveform and indicates errors.

Compared Data, Captured Data, and Errored Data all use the Error State Display for presentation. You can switch between the three views within the window. Alternatively, you can open the window multiple times to inspect the contents concurrently.

Bit Error Rate Display, Error State Display, and Waveform Viewer are accessible by clicking buttons in the tool bar.

Operating the User Interface

The preferred instrument for operating the system is the mouse or the touchpad.

How to Use the Mouse or Touchpad

Some areas of the windows are “active” areas. They can be identified by the cursor changing its shape when placed over these areas.

If you press the right mouse button on an active area, a context menu pops up that lists the actions you can perform on the chosen item. If there is a default action defined for a context menu, this is indicated with bold text.

The default action is automatically executed if you double-click on an active area with the left mouse button.

NOTE If you have highlighted an item by clicking on it once with the left mouse button, you can also select the available actions from the menus of the menu bar on top of the window frame.

How to Navigate With the Keyboard

There are people who hate mice. They may use the keyboard instead:

- Tab and Shift+Tab move the cursor.
- Shift+F10 opens the context menu.
- To close the active window press Ctrl+F4.
- To switch between open windows press Ctrl+F6.
- To terminate the system run press Alt+F4.

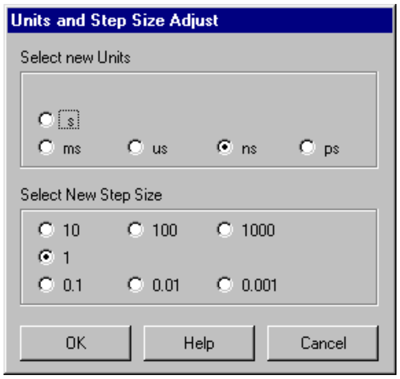
If you type something into a data entry field, terminate your input with Enter. This causes the software to check the input and react.

How to Change Units and / or Vernier Steps

Some windows contain data entry fields with vernier buttons and units.

To change the default unit or vernier step size:

1 Click on the unit button to open the *Units and Step Size Adjust* window.



Units and Step Size Adjust

Select new Units

○ s
○ ms ○ us ● ns ○ ps

Select New Step Size

○ 10 ○ 100 ○ 1000
● 1
○ 0.1 ○ 0.01 ○ 0.001

Figure 66 Units and Step Size Adjust Window

2 Change the unit and step size as desired.

3 Click OK.

NOTE You can also click with the right mouse button on the unit of a data entry field and change unit and step size from the context menu.

How to Use the Window Selection Box

This dialog box appears when you are opening the Parameter Editor or the Waveform Viewer from the respective menu item or icon in the main window. It appears also, if you open the Error State Display for a device that has more than one output port.

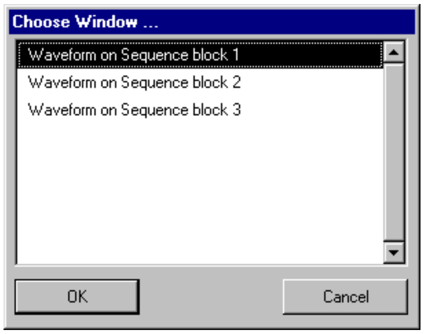

Choose Window ...
Waveform on Sequence block 1
Waveform on Sequence block 2
Waveform on Sequence block 3
OK
Cancel

Figure 67 Window Selection Dialog Box Example

1. Click on an item in this dialog box to select this item to be presented in the respective editor/viewer.
2. Click OK.

   The respective editor/viewer opens.

For information on the usage of the Parameter Editor, see *“How to Start the Parameter Editor for Global Parameters” on page 176* or *“How to Start the Parameter Editor for Ports/Channels” on page 224*.

For information on the Error State Display, see *“How to View Captured Test Results” on page 350*.

For information on the Waveform Viewer, see *“How to View Waveforms” on page 356*.

Items of the Main Window

The main window of the Agilent 81250 Graphical User Interface provides easy access to and control of all features and functions of the system.

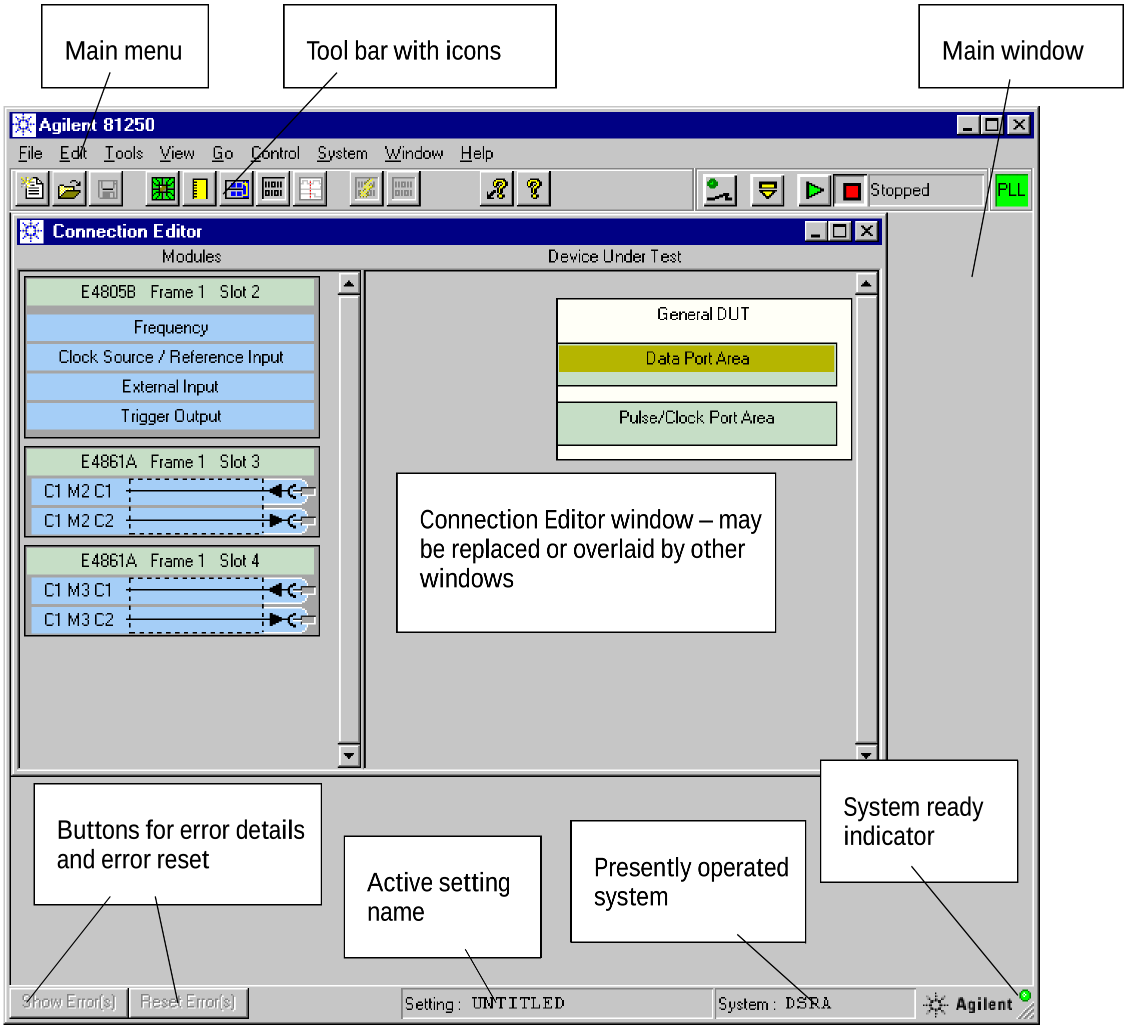


Figure 68 **Agilent 81250 Main Window**

The components of the main window are explained below.

Main menu The main menu is located at the top of the main window.

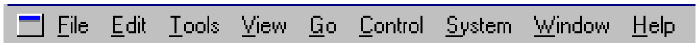


Figure 69 **Agilent 81250 Main Menu**

It provides access to the following menus:

"File Menu" on page 153

"Edit Menu" on page 159

"Tools Menu" on page 163

"View Menu" on page 164

"Go Menu" on page 166

"Control Menu" on page 168

"System Menu" on page 169

"Window Menu" on page 171

"Help Menu" on page 172

Tool Bar

In the main window there is also a tool bar located at the top, which contains icon buttons providing shortcuts to windows and dialogs. They are explained along with the respective menus.

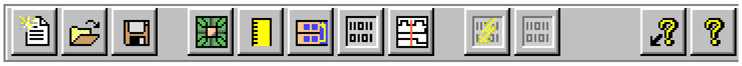


Figure 70 Tool Bar

NOTE

Not all items of the menus and the tool bar are available at all times. Whether an item is available or not depends on the active window and the current situation.

For example, the error state display is not available, if you have decided to perform a bit error rate measurement, and vice versa.

The individual windows therefore provide context menus which can be opened by positioning the cursor to the area of interest and clicking the right mouse button.

Run Control Area

In the upper right-hand corner of the main window you see the Run Control Area. Here you find a set of icon buttons and a status field showing the current test status.

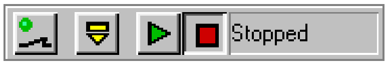


Figure 71 Run Control Area

For an explanation of these buttons see *"Control Menu" on page 168*.

PLL Lock Indicator

When an external clock source is used, the phase locked loop of the master clock module locks onto that source. In case the clock system could not lock or has lost its synchronization, the PLL lock indicator turns from green to red.

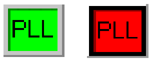


Figure 72 **PLL Locked / Unlocked Indicator**

If this happens while a test is runnung, the test is invalid.

At a ParBERT 43G DEMUX system, the PLL lock indicator can also appear as shown below:

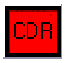
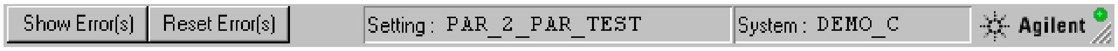


Figure 73 **CDR Unlocked Indicator**

If this happens, then the clock data recovery (CDR) circuit of the E4869A DEMUX module could not synchronize on the incoming data stream. As the generated clock signal is undefined, it is useless to continue the test. You may consider using an external clock.

Status Line

At the bottom of the main window you can see the status line.



Setting : PAR_2_PAR_TEST

System : DEMO_C

Figure 74 **Status Line**

Located in the status line are:

- The *Show Error(s)* and the *Reset Error(s)* buttons.

  If you enter invalid data into a text field of an editor window or a dialog box, the software reports an error by highlighting all erroneous fields and the *Show Error(s)* button.

  Clicking this button opens an error window that explains the error. Clicking the *Reset Error(s)* button resets the invalid entry to the last valid value.

- The name of the loaded test setting.
- The name of the presently operated system.

File Menu

The *File* menu is primarily used for storing data in files and retrieving data from files.

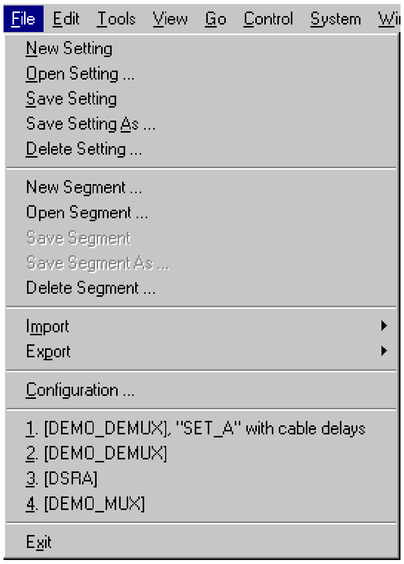


Figure 75 File Menu

New Setting

Closes the current setting and returns you to the default setting. The default setting is used for starting with a new device under test (DUT).

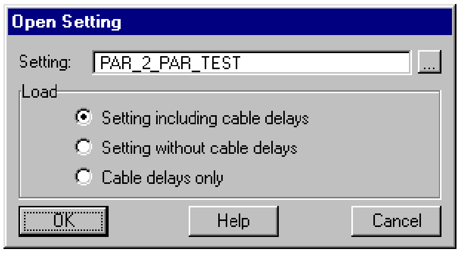
If the current setting has been changed and not saved, you will be asked whether you wish to save the current setting before the default setting is loaded. The current setting is indicated in the bottom line of the window frame.

Shortcut: New Setting icon in the tool bar.

Open Setting

Used to load one of the saved settings.

If the current setting has been changed and not saved, you will be asked whether you wish to save the current setting before the new setting is loaded. The current setting is indicated in the bottom line of the window frame.



Open Setting

Setting: PAR_2_PAR_TEST

Load

- ● Setting including cable delays
- ○ Setting without cable delays
- ○ Cable delays only

Figure 76 Open Setting Dialog

Default is the last opened setting. Others can be chosen from the list. The options for opening a setting are:

- Complete setting including cable delays
- Setting without cable delays (e.g. if connection cables to the DUT have been changed)
- Cable delays only (e.g. if the same cables are used for connecting to a new DUT)

Shortcut: Open Setting icon in the tool bar.

Save Setting

Saves the currently loaded setting on the disk, including all changes that have been made. Overwrites the previous version.

For saving or creating a new setting or keeping the original setting use *Save Setting As* instead.

Shortcut: Save Setting icon in the tool bar.

Save Setting As

Displays a list of the stored settings and enables you to overwrite one of these or to save the current setting under a new name on the disk.

Delete Setting

Displays a list of the stored settings and enables you to delete one or several of these.

New Segment

Enables you to create a new data segment. You need to select a pool (global or local) and enter a segment name. You can change width and length.

After that you enter the Segment Editor see *“Creating and Editing Segments” on page 309*.

The option *New Segment* can also be invoked from the Standard or Detail Mode Sequence Editor.

Open Segment

Displays a list of the stored segments. After choosing a segment, you enter the Segment Editor for investigating or changing the properties of the chosen segment.

Shortcut: Segment Editor icon in the tool bar.

Save Segment

Only available if the Segment Editor is active and the segment has been changed.

Saves the modified segment under its original name.

Shortcut: If the segment has been changed and you terminate the Segment Editor, you will be asked whether you wish to save your changes.

Save Segment As

Only available if the Segment Editor is active.

Displays a list of the stored segments. The list can be toggled to show the contents of the LocalSegments or GlobalSegments pool. Enables you to overwrite one of the existing segments or to save the current segment under a new name on the disk.

Delete Segment

Displays a list of the stored segments. The list can be toggled to show the contents of the LocalSegments or GlobalSegments pool. Enables you to delete one or several of the segments.

Import

Enables you to import settings or segments that have been exported to files.

Import Setting

Importing a setting replaces the current setting.

If the current setting has been changed and not saved, you will be asked whether you wish to save the current setting before the new setting is imported.

The file containing the setting can be found with the browser.

The imported setting can then be saved with the *Save Setting As* function.

Import Segments

Segments can be imported into the local or global segments pool.

The file containing the segments can be found with the browser.

The file can include more than one segment. You can therefore specify whether you wish to overwrite existing segments or not.

Export

Enables you to export settings or segments as files.

The destination path can be specified with the browser. Enter a new file name or select a file to overwrite.

Export Setting

When exporting a setting, the options are:

- Complete setting including cable delays
- Setting without cable delays (e.g. if connection cables to the DUT have been changed)
- Cable delays only (e.g. if the same cables are used for connecting to a new DUT)

The generated file is an ASCII file which can be investigated with a suitable text editor. A setting file contains all the firmware commands that establish that setting.

Export Segments

When exporting segments, the options are:

- all from GlobalSegments pool
- all from LocalSegments pool
- a single segment which can be chosen from the browser

The generated file is an ASCII file which can be investigated with a suitable text editor. Segment files are Vector Format text files.

Configuration

You may wish to operate a different ParBERT system.

You may also may wish to start and use two user interfaces in order to operate two ParBERT systems in parallel.

So, every user interface has to be configured.

This menu item enables you change the characteristics of the present user interface. It opens the User Interface Configuration dialog. For details see *"How to Configure the User Interface" on page 140*.

For system configurations please refer to the *Agilent 81250 Installation Guide*.

List of User Interface Configurations

The *File* menu includes a list of the most recent user interface configurations. This makes it easy to change the currently operated system against another one.

If you wish to operate a different ParBERT system from your present user interface, just choose from the list.

The following actions lead to a new entry:

- Closing the Agilent 81250 Graphical User Interface Configuration dialog with OK
- New Setting
- Open Setting
- Save Setting
- Save Setting As

The new entry is put in position 1. The numbers of the other entries are incremented by 1. If the entry is identical to another entry in the list, this entry is removed. The list keeps up to four entries.

The list of the recently used startup configurations is shared among all user interfaces, independent on the startup parameter set they use.

Exit

Closes the user interface and terminates the system run.

Shortcut: The Close button in the upper right-hand corner of the window frame or Alt+F4.

Edit Menu

The *Edit* menu is used for preparing a test. It contains the editing functions that can be used in the various test setup editors.

The appropriate menu items are enabled when the respective test setup editor window is active.

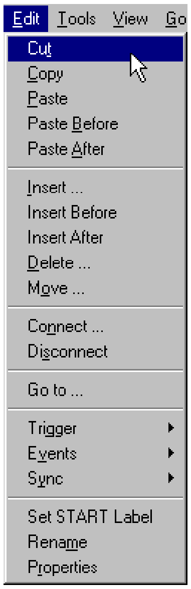


Figure 77 Edit Menu

Cut

Copies the highlighted item to the clipboard and removes it from the current window.

Shortcut: Ctrl+x

Copy

Copies the highlighted item to the clipboard.

Shortcut: Ctrl+c

Paste

Only available if data has been copied to the clipboard.

Inserts the clipboard contents at the specified location.

Shortcut: Ctrl+v

Paste before

An option of the Detail Mode and Data/Sequence Editors. Only available if a block has been copied to the clipboard.

Inserts the clipboard contents above the currently highlighted block.

Paste after

An option of the Detail Mode and Data/Sequence Editors. Only available if a block has been copied to the clipboard.

Inserts the clipboard contents below the currently highlighted block.

Insert

An option of the Connection Editor and the Segment Editor.

Inserts a new port, terminal, vector or trace at the specified location.

Insert before

An option of the Detail Mode and Data/Sequence Editors.

Inserts a new block above the currently highlighted block.

Insert after

An option of the Detail Mode and Data/Sequence Editors.

Inserts a new block below the currently highlighted block.

Delete

An option of the Connection Editor and the Segment Editor.

Deletes the highlighted object.

Move

An option of the Connection Editor.

Moves the highlighted terminal up or down.

Shortcut: Drag and drop.

Connect

An option of the Connection Editor.

Enables you to connect the highlighted terminal to a frontend connector.

Shortcut: Drag and drop.

Disconnect

An option of the Connection Editor.

Enables you to disconnect terminals from frontend connectors.

Shortcut: Drag the connection (i.e. the blue rectangle next to the terminal, labeled Cx My Cz) to the left somewhere into the modules area but not on a connector and release the mouse button.

Go to

An option of the Segment Editor and the Error State Display.

Enables you to jump to a specified vector address.

Find

An option of the Segment Editor.

Enables you to locate certain patterns in a memory-type data segment.

Trigger

An option of the Detail Mode and Data/Sequence Editors.

Used to enable or disable the generation of a trigger pulse associated with a block.

Events

An option of the Detail Mode and Data/Sequence Editors.

Enables you to specify events and define actions upon events (such as sequence branches). See *“How to Specify Events and Reactions Upon Events” on page 298.*

Sync

An option of the Detail Mode Sequence Editor.

Enables you to activate, disable, or change the automatic sampling point adjustment of the analyzer frontends connected to a DUT output port (see *"How to Synchronize an Analyzer With Incoming Data" on page 272*).

Provides also the access to the DEMUX rewiring feature (see *"How to Synchronize an Analyzer With Incoming Data" on page 272*).

Set Start

An option of the Sequence Editors.

Enables you to specify a start block of a sequence. When running the test, the start block will be the first to be executed. Any blocks above the start block are ignored when the test is executed. This option is useful to determine different entry points in a data sequence without the need of major modifications in the sequence content itself.

NOTE The start block is automatically named START. In order to avoid confusion, you should not assign that label manually to a block.

Rename

An option of the Connection Editor.

Enables you to rename the highlighted port or terminal of the DUT.

Properties

An option of several editors.

Shows the individual properties (parameters) of the highlighted object.

Tools Menu

The *Tools* menu contains additional options of the Segment Editor.

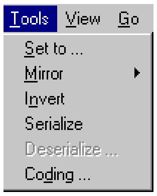


Figure 78 Tools Menu

Set to

With this option you can set a highlighted section in the Segment Editor to a specified value.

Mirror

With this option you can mirror a highlighted section in the Segment Editor either horizontally or vertically.

Invert

With this option you can invert a highlighted section in the Segment Editor—zeros to ones, ones to zeros.

Serialize

With this option of the Segment Editor you can convert a parallel memory segment holding multiple traces into a serial segment holding just one trace.

Deserialize

With this option of the Segment Editor you can convert a serial memory segment holding one trace into a parallel segment holding multiple traces.

Coding

With this option of the Segment Editor you can change the coding of the data segment. Selecting this option opens the Data Converter window.

View Menu

The options of the *View* menu allow you to change the appearance of data and waveform displays and to choose a suitable result display.

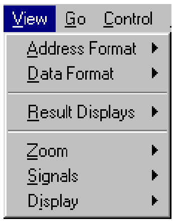


Figure 79 View Menu

Address Format

Enables you to change the displayed address format. Options are:

- Decimal
- Hexadecimal
- Octal

Data Format

Enables you to change the displayed data format. Options are:

- Binary
- Hexadecimal
- Octal

Additional options are available if the Segment Editor has been started from the Sequence Editor for editing or creating a segment within a block:

- Trace View: Shows trace numbers of the data segment
- Port View: Shows the data port to which the segment is assigned
- Terminal View: Shows the terminal names of the data port
- Connector View: Shows the identifications of the connected channels

Result Display

Enables you to choose a suitable display. Available options depend on the kind of test that has been specified in the Measurement Configuration window:

- Bit Error Rate—only available, if "Error Rate Measurement" has been selected. Opens the Bit Error Rate display window.

  Shortcut: Bit Error Rate Display icon in the tool bar.

- Compared Data—only available, if "Compare and Capture" or "Compare and Acquire around Error" has been selected. Opens the Error State display window.

  Shortcut: Error State Display icon in the tool bar.

- Captured Data—not available, if "Error Rate Measurement" has been selected. Opens the Error State display window.

  Shortcut: Error State Display icon in the tool bar.

- Errored Data—only available, if "Compare and Capture" or "Compare and Acquire around Error" has been selected. Opens the Error State display window.

  Shortcut: Error State Display icon in the tool bar.

- Waveform—not available, if "Error Rate Measurement" has been selected. Opens the Waveform Viewer.

  Shortcut: Waveform Viewer icon in the tool bar.

NOTE Compared Data, Captured Data, and Errored Data all use the Error State Display for presentation. You can switch between the three views within the window. Alternatively, you can open the window multiple times to inspect the contents concurrently.

Zoom

An option of the Waveform Viewer.

Enables you to zoom into or out of the waveform display.

Signals

An option of the Waveform Viewer.

Enables you to change the waveform amplitudes in the display and to rearrange the signal order.

Display

An option of the Waveform Viewer.

Determines the display unit for the Waveform Viewer—either time (for DUT input ports) or number of samples (for output ports).

Go Menu

The options of the *Go* menu can be used to open the various test setup editors. They provide also access to functions which are not represented by tool bar buttons.

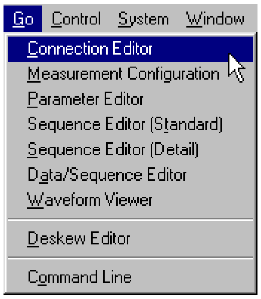


Figure 80 Go Menu

Connection Editor

Opens the Connection Editor. For details see *"Connecting the DUT" on page 195*.

Shortcut: Connection Editor icon in the tool bar.

Measurement Configuration

Opens the Measurement Configuration window. For details see *"Choosing the Kind of Measurement" on page 261*.

Shortcut: Measurement Configuration icon in the tool bar.

Parameter Editor

Opens the Parameter Editor. For details see *"How to Start the Parameter Editor for Global Parameters" on page 176* and *"How to Start the Parameter Editor for Ports/Channels" on page 224*.

Sequence Editor (Standard)

Opens the Standard Mode Sequence Editor, if applicable. If not, the Detail Mode Sequence Editor is started.

For details see *"Creating the Stream of Generated and Expected Data" on page 265*.

Shortcut: Sequence Editor icon in the tool bar.

Sequence Editor (Detail)

Opens the Detail Mode Sequence Editor.

This editor is always available. It is used for specifying the sequence of data blocks which reference the generated and expected data segments. The sequence can be straightforward or contain loops. See *"The Detail Mode Sequence Editor" on page 286*.

Shortcut: Sequence Editor icon in the tool bar.

Data / Sequence Editor

Opens the Data/Sequence Editor (see *"How to Start the Data/Sequence Editor" on page 336*).

Waveform Viewer

Opens the Waveform Viewer (see *"How to View Waveforms" on page 356*).

Shortcut: Waveform Viewer icon in the tool bar.

Deskew Editor

Opens the Deskew Editor. For details see *"How to Compensate for Internal and External Delays" on page 364*.

Command Line

Opens the Command Line input window. For details see *"How to Execute Firmware Commands" on page 382*.

Control Menu

The items in the *Control* menu are used to control the actual test run on the DUT.

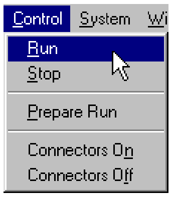


Figure 81 Control Menu

All control actions of this menu can be performed by clicking the respective buttons in the upper right-hand corner of the main window. The current status of the test is also displayed at the right-hand side of these buttons.

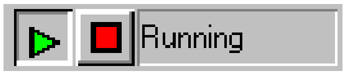
Run

Starts a test.

Shortcut: Run button in the tool bar.

Stop

Stops a running test.



Shortcut: Stop button in the tool bar.

Prepare Run

Downloads the test sequence to the modules. This is required for immediate reaction on a start trigger.

Shortcut: Prepare button of the tool bar.

Connectors On / Off

Switches relays inside the frontends:

- *Connectors Off* opens all input and output relays. This isolates the DUT electrically from the system.

  For channels with data rates above 675 Gbit/s, this behavior can be changed. It is possible to specify whether the relays shall be switched or whether the frontends shall be disconnected by grounding.

For details see: *"How to Set the Global Disconnect Mode" on page 239*.

- *Connectors On* connects all module connectors with activated outputs or inputs electrically with the cables. Connector outputs or inputs can be activated or deactivated with the Parameter Editor.

Shortcut: Connectors On/Off button in the tool bar.

System Menu

The *System* menu has functions for testing the system's integrity. These tests can be performed at any time, as long as no test is running.

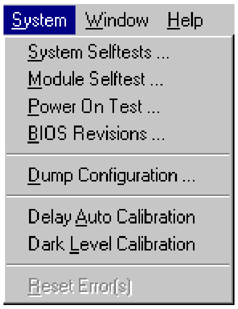


Figure 82 System Menu

Note that the optional diagnostics software package provides additional tests which in case of problems can identify defective field replaceable units.

System Selftests

Provides a window, from which the complete selftest or subsets can be started. Ensures that all modules respond. Returns the current firmware revisions of the modules and the identification numbers of installed frontends.

Module Selftest

Enables you to check all or single modules. Checks the frontends built into the modules. For technical reasons, this test may take a minute.

Power On Test

This test is automatically performed at power on. Checks all modules.

BIOS Revisions

Returns the current firmware revisions of the modules.

Dump Configuration

Enables you to write the current system configuration to a file. You have to specify the path and file name.

The file will contain a contiguous string of ASCII characters, showing the identifications of all modules and frontends.

Delay Auto Calibration

If any frontends or modules have been replaced or added to the system, you have to perform the *Delay Auto Calibration*. This aligns the built-in timing circuits and establishes the internal zero delay. The *phase accuracy* of the modules and frontends given in the *Technical Specifications* refers to this delay.

The voltage-controlled delay circuits are susceptible to temperature changes. You should therefore also run the *Delay Auto Calibration* if your environment temperature differs continually by more than ±10 °C from the temperature of the last calibration (±5 °C for the E4861A modules).

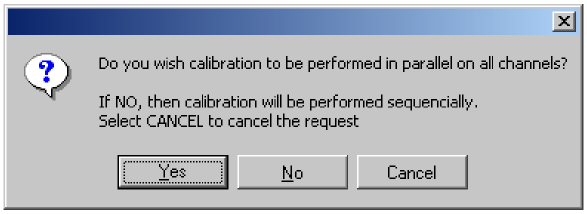
NOTE *Delay Auto Calibration* has to be run after a warm-up period of at least 30 minutes and **before** the zero adjust and cable delay compensation procedures are performed.

Depending on the complexity of the system, *Delay Auto Calibration* can take several minutes.

Dark Level Calibration

Dark Level Calibration allows you to calibrate all optical analyzers in one go. This requires that you disconnect the optical fibers and install the plastic caps that shield the connectors from light (for details see *"How to Set Optical Analyzer Levels" on page 248*).

If your system includes more than one optical analyzer, you can choose between parallel and sequential calibration.



Do you wish calibration to be performed in parallel on all channels?

If NO, then calibration will be performed sequencially.
Select CANCEL to cancel the request

Figure 83 **Choice of Dark Level Calibration Procedures**

Parallel calibration requires that all optical analyzer input connectors are covered.

The sequential procedure moves automatically from one optical analyzer to the next. It stops before each measurement, so that you can attach the cap on the optical connector that is going to be calibrated, or skip that connector.

Reset Error(s)

This menu item corresponds to the *Reset Error(s)* button in the lower left-hand corner of the main window. If invalid data was entered in any of the editor windows (e.g. the Parameter Editor), the software reports this error by highlighting all erroneous fields and the *Show Error(s)* button.

Reset Error(s) then resets the invalid input to the last correct value.

Window Menu

The *Window* menu contains standard functions provided by Microsoft Windows. You can use these items to rearrange the open windows or to switch between them.

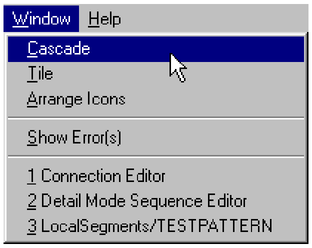


Figure 84 **Window Menu**

Shortcut for switching between windows: Ctrl+F6

Cascade

Superimposes all open windows. You can click any window title to bring that window to the front.

Tile

Arranges all open windows within the window frame.

Show Error(s)

This menu item corresponds to the *Show Error(s)* button in the lower left-hand corner of the main window. If invalid data was entered in any of the editor windows (e.g. Parameter Editor), the software reports this error by highlighting all erroneous fields and the *Show Error(s)* button.

Show Error(s) then displays an error window describing the error in detail.

Help Menu

The *Help* menu is supposed to be self-explanatory. You can start with the table of contents or search from the alphabetical index.

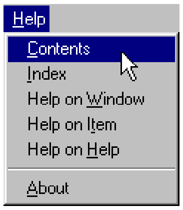


Figure 85 Help Menu

The options of the *Help* menu are:

- Contents.

  This opens the Help Desk window presenting an introduction to the system.

- Index.

  This opens the Help Desk window presenting an introduction to the system.

- Help on Window.

  This opens the Help Desk window presenting information on the currently active window.

- Help on Item.

  This option corresponds to the Help on Item button in the tool bar. When selected, the cursor changes its shape.

  Place the cursor on the item of interest an click the left mouse button. The Help Desk window presents information on the selected item.

- Help on Help.

  This opens the Help Desk window presenting an explanation how to use help.

- About.

  This opens a window telling the software version and contact and copyright information.

TIP For help on a specific item place the cursor on this item and press F1. The Help Desk window then presents information on this item.

Setting Global System Parameters

Global system parameters refer to the master clock module. If slaves are connected, they will follow the master.

The clock module has

- a built-in 10 MHz reference,
- a PLL-controlled oscillator,
- pulse delay circuits,
- a frequency multiplier/divider,
- two input connectors named CLOCK / REF INPUT and EXT INPUT,
- one output connector named TRIGGER OUTPUT.

The Parameter Editor is used for setting appropriate parameters.

This chapter provides instructions on how to set up the clock module:

- *"How to Start the Parameter Editor for Global Parameters" on page 176*
- *"How to Set the Clock Frequency" on page 177*
- *"How to Choose the Clock Source" on page 187*
- *"How to Set the Characteristics of the External Input" on page 190*
- *"How to Set the Characteristics of the Trigger Output" on page 192*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *"Easy Frequency Setting"*.

How to Start the Parameter Editor for Global Parameters

There are global and channel-related system parameters. Both are set with the Parameter Editor. To support all kinds of parameters for the central clock module, data generator/analyzer modules, channels, ports, terminals and so on, the Parameter Editor has several modes of operation.

The global system parameters comprise the setup of:

- System, port, and channel frequencies,
- Clock source,
- External input,
- Trigger output.

The simplest way to access these setups is from the Connection Editor:

1 Double-click the corresponding fields in the *Modules* section.

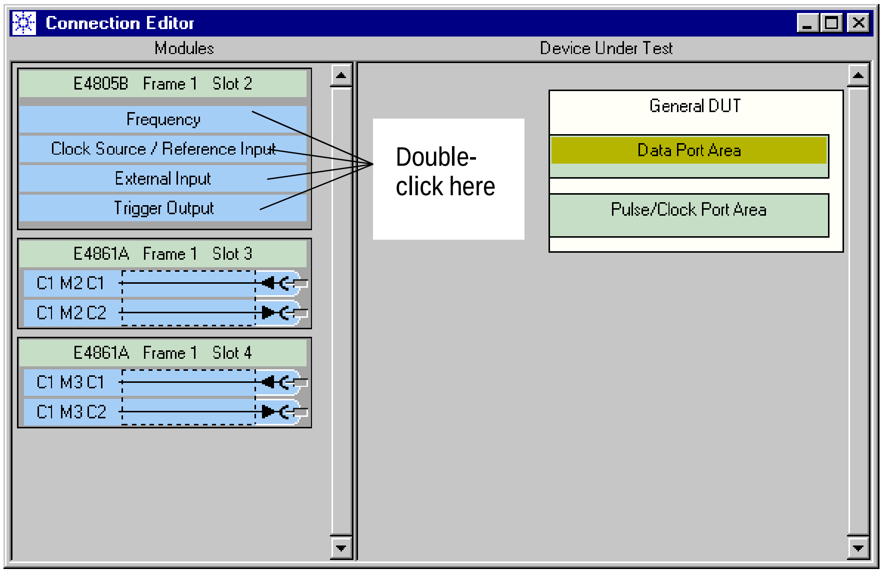


Figure 86 Accessing Global Parameters

If the Connection Editor is not displayed:

1 Choose *Parameter Editor* from the *Go* menu.

You get a list of all the items that have been configured and can have parameters. As long as you have not connected any DUT terminals with the Connection Editor, the list starts with the clock module.

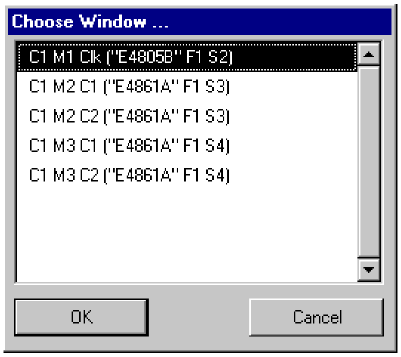

Choose Window ...
C1 M1 Clk ("E4805B" F1 S2)
C1 M2 C1 ("E4861A" F1 S3)
C1 M2 C2 ("E4861A" F1 S3)
C1 M3 C1 ("E4861A" F1 S4)
C1 M3 C2 ("E4861A" F1 S4)
OK
Cancel

Figure 87 Parameter Editor Selection Window

2 Select the master clock module ("C1 M1 Clk") and click *OK*.

The Frequency setup window appears.

NOTE Once the Parameter Editor has been started, it provides on the top of the window a browser labeled *Resource* and up/down arrows that enable you to switch to another item.

How to Set the Clock Frequency

You have to set a general system clock frequency. Then you may set individual clock frequencies for:

- Connected DUT data ports
- Connected pulse port terminals
- Unconnected but enabled instrument channels

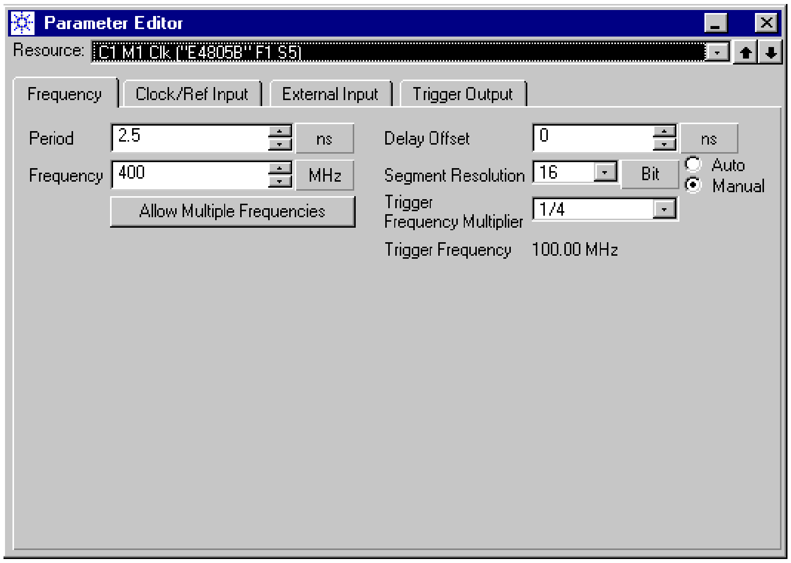
Individual clock frequencies can be generated by multiplying or dividing the system clock frequency by factors of 2. For details see:

"How to Set the General System Frequency" on page 178

"How to Use Multiple Frequencies" on page 182

How to Set the General System Frequency

1 Open the Parameter Editor for the clock module (see *“How to Start the Parameter Editor for Global Parameters” on page 176*). Choose the *Frequency* page.



Resource: C1 M1 Clk ("E4805B" F1 S5)

Period 2.5 ns

Frequency 400 MHz

Delay Offset 0 ns

Segment Resolution 16 Bit ○ Auto ● Manual

Trigger Frequency Multiplier 1/4

Trigger Frequency 100.00 MHz

Figure 88 Setting the General System Frequency

NOTE The *Frequency* page shows more details if your system configuration requires multiple frequencies.

You can specify the desired *Period* or the desired *Frequency*. Both are equivalent. The period is always the reciprocal of the frequency, and vice versa.

2 Accept or change the units.

The Parameter Editor displays default units (ns/MHz) and has default vernier steps. Both can be changed (see *“How to Change Units and/or Vernier Steps” on page 148*).

3 Decide on the general *Segment Resolution* adjustment mode.

Automatic adjustment is recommended. In this mode, the program calculates a suitable general segment resolution. In manual mode, you have to adjust it to an adequate value (see below).

4 Decide on using the TRIGGER OUTPUT of the clock module.

The TRIGGER OUTPUT can be used to generate a continuous clock signal or single pulses upon events. Its mode can be set on the *Trigger Output* page.

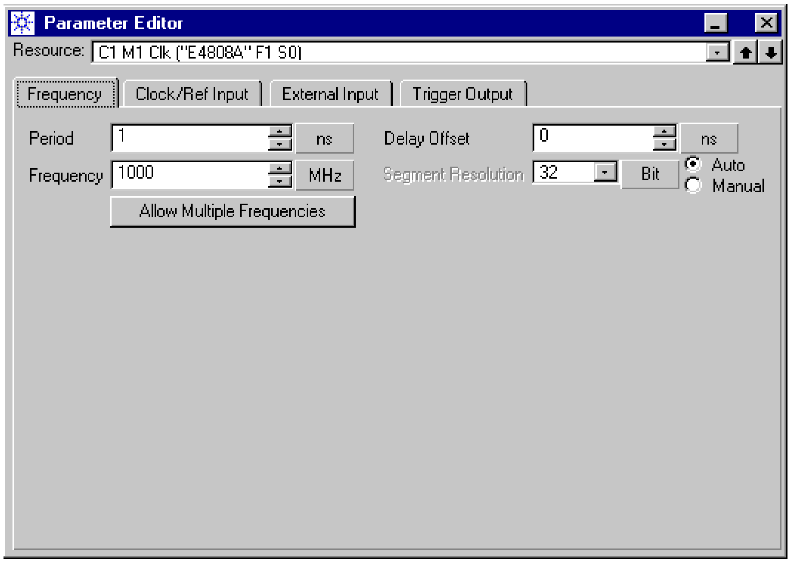
If you are not going to use the TRIGGER OUTPUT of the clock module for generating a clock, put it into *Sequencer* mode (see *"How to Set the Characteristics of the Trigger Output" on page 192*).

When the TRIGGER OUTPUT is in *Sequencer* mode, its frequency is neither displayed nor checked. You do not have to worry about the *Trigger Frequency Multiplier* setting.

5 Set the desired general *Frequency* or *Period.*

If you have typed a number, terminate your input with the Enter or Return key. This updates the other field (*Period* or *Frequency*).

Your window may look like the following:



Resource: C1 M1 Clk ("E4808A" F1 S0)

Period 1 ns

Frequency 1000 MHz

Delay Offset 0 ns

Segment Resolution 32 Bit ● Auto ○ Manual

Figure 89 Frequency Setting with Automatic Segment Resolution Adjustment

6 If desired, set a global *Delay Offset.*

A global delay offset enables you to specify later-on negative time offsets for individual signals. Such signals then appear in advance of the regular clock pulse.

7 If you have enabled manual *Segment Resolution* adjustment, check its value.

The segment resolution has an impact on the available memory resources for generating or capturing data. The default setting of the segment resolution is 4 bits for E4832A modules and 16 bits for E4861A/B modules.

The segment resolution is coupled with a certain Frequency Multiplier (FM) factor. The available multiplication factors that can be used for multiplying the system clock are called Frequency Multiplier Range (FMR).

The following tables show the interdependencies.

Table 12 Clock Rates, Segment Resolution, Memory Depth for E4832A Modules

System Clock Frequency Mbit/s	Segment Resolution bits	Memory Depth bits	Frequency Multiplier Range
≤ 42.1875	1	131,008	1, 2, 4, 8, 16
≤ 84.375	2	262,016	1/2, 1, 2, 4, 8
≤ 168.750	4	524,032	1/4, 1/2, 1, 2, 4
≤ 337.500	8	1,048,064	1/8, 1/4, 1/2, 1, 2
≤ 675.000	16	2,097,152	1/16, 1/8, 1/4, 1/2, 1

Table 13 Clock Rates, Segment Resolution, Memory Depth for E4861A Modules

System Clock Frequency Mbit/s	Segment Resolution bits	Memory Depth bits	Frequency Multiplier Range
333.334 to 675.000	16	2,097,152	1, 2, 4
≤ 1,350.000	32	4,194,304	1/2, 1, 2
≤ 2,700.000	64	8,388,608	1/4, 1/2, 1

Table 14 Clock Rates, Segment Resolution, Memory Depth for E4810A, E4811A, and E4861B Modules

System Clock Frequency Mbit/s	Segment Resolution bits	Memory Depth bits	Frequency Multiplier Range
20.834 to 41.666	1	131,072	1, 2, 4, 8, 16, 32, 64, 128
≤ 82.333	2	262,144	1/2, 1, 2, 4, 8, 16, 32, 64
≤ 166.666	4	524,288	1/4, 1/2, 1, 2, 4, 8, 16, 32
≤ 333.333	8	1,048,576	1/8, 1/4, 1/2, 1, 2, 4, 8, 16
≤ 666.666	16	2,097,152	1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8
≤ 1,333.333	32	4,194,304	1/32, 1/16, 1/8, 1/4, 1/2, 1, 2, 4
≤ 2,700.000	64	8,388,608	1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, 2
≤ 3,350.000	128	16,777,216	1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1

NOTE Depending on the desired clock rate, it is possible to choose a lower or higher segment resolution than displayed, as shown in the tables above. If this is done for the analyzer frontends of an E4832A module, then the delay vernier and the functions for automatic sampling point adjustment are not available.

The 10.8 Gbit/s modules have just one frequency range:

Table 15 Clock Rate, Segment Resolution, Memory Depth for E4866A / E4867A Modules

System Clock Frequency Gbit / s	Segment Resolution bits	Memory Depth bits	Frequency Multiplier Range
9.5 to 10.8	256	33,554,432	1

For details please refer also to *"Frequency Multiplier and Segment Resolution" on page 68*.

NOTE Not all frequencies and hence frequency multiplier factors are supported by all modules and frontends. The system will report an error, if the chosen segment resolution or frequency multiplier factor does not fit to **all** the components. In this case you may wish to change the setup individually for some channels and use multiple frequencies.

8 If a *Trigger Frequency* is displayed, check the *Trigger Frequency Multiplier* value.

The *Trigger Frequency Multiplier* refers to the TRIGGER OUTPUT of the clock module. The TRIGGER OUTPUT can be used to generate a continuous clock signal or single pulses.

By default, the trigger clock frequency equals the master clock frequency. By setting the *Trigger Frequency Multiplier* to a factor other than 1, you can change the trigger clock frequency. The factor can only be set if the TRIGGER OUTPUT is in *Clock Generator* mode.

NOTE The master clock frequency is limited to 675 MHz. If you have set a system clock frequency above 675 MHz, you have to adjust the *Trigger Frequency Multiplier*.

For example, if you have chosen a system clock frequency of 2.7 GHz, the maximum *Trigger Frequency Multiplier* setting is 1/4th.

See also *"How to Set the Characteristics of the Trigger Output" on page 192*.

How to Use Multiple Frequencies

Multiple frequencies can be required by the DUT, if some ports or pins are operated at different clock rates.

Multiple frequencies are also required if your ParBERT system is equipped with frontends which cannot be operated under one and the same frequency setup.

Additional clock frequencies can be generated by multiplying or dividing the system clock frequency by factors which are multiples of 2.

ParBERT 43G system

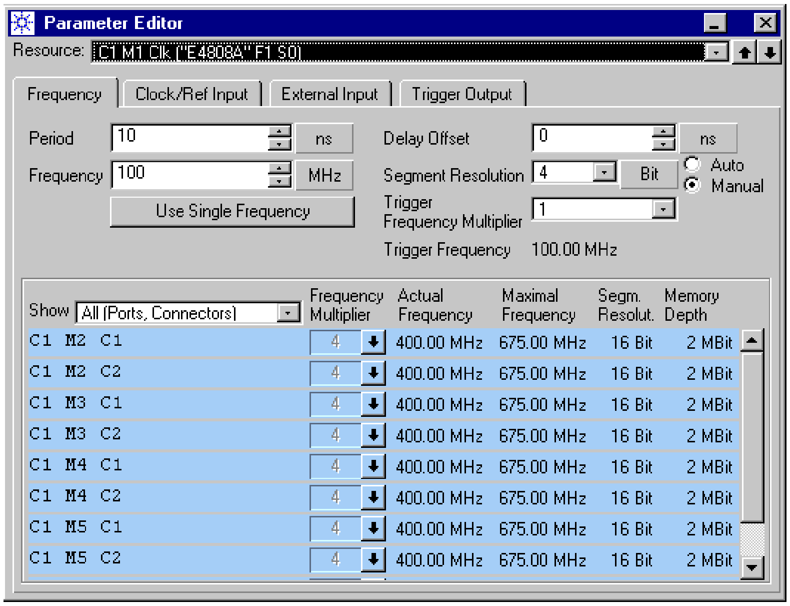
A ParBERT 43G system by default uses multiple frequencies: 37 GHz to 43.2 GHz for the MUX/DEMUX module and 1/16th for the data generator/analyzer modules.

Setting Multiple Frequencies

1 Open the Parameter Editor for the clock module (see *“How to Start the Parameter Editor for Global Parameters” on page 176*). Choose the *Frequency* page.

2 Click *Allow Multiple Frequencies.*

As long as you have not connected any DUT terminals to frontends with the Connection Editor, the list shows all the connectors:



Resource: C1 M1 Clk ("E4808A" F1 S0)

Period 10 ns

Frequency 100 MHz

Delay Offset 0 ns

Segment Resolution 4 Bit ○ Auto ● Manual

Trigger Frequency Multiplier 1

Trigger Frequency 100.00 MHz

Show All (Ports, Connectors)

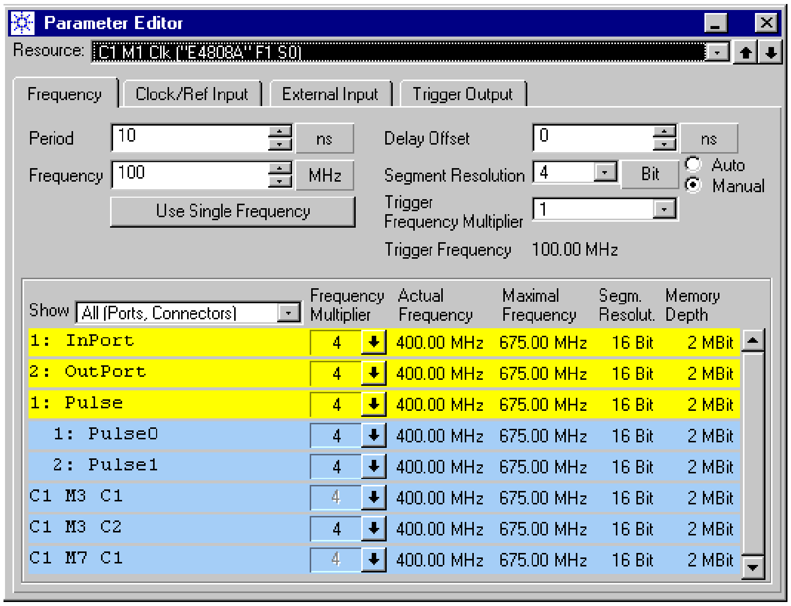
	Frequency Multiplier	Actual Frequency	Maximal Frequency	Segm. Resolut.	Memory Depth
C1 M2 C1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M2 C2	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M3 C1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M3 C2	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M4 C1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M4 C2	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M5 C1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M5 C2	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit

Figure 90 Setting Multiple Frequencies—No Ports Defined

NOTE A list like this appears automatically if your system is composed of modules which require a multiple frequency setup.

The window shows default values. The *Frequency Multipliers* are grey and cannot be changed. The *Segment Resolution* is set to manual adjustment. It would be set to automatic if the system included modules that support only one segment resolution (like the 10.8G data modules).

If the DUT terminals have already been connected to frontends with the Connection Editor, the list shows all the data ports, pulse port terminals, and unconnected channels, as illustrated below:



Resource: C1 M1 Clk ("E4808A" F1 S0)

Period 10 ns

Frequency 100 MHz

Delay Offset 0 ns

Segment Resolution 4 Bit ○ Auto ● Manual

Trigger Frequency Multiplier 1

Trigger Frequency 100.00 MHz

Show All (Ports, Connectors)

	Frequency Multiplier	Actual Frequency	Maximal Frequency	Segm. Resolut.	Memory Depth
1: InPort	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
2: OutPort	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
1: Pulse	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
1: Pulse0	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
2: Pulse1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M3 C1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M3 C2	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit
C1 M7 C1	4	400.00 MHz	675.00 MHz	16 Bit	2 MBit

Figure 91 Setting Multiple Frequencies for Connected and Unconnected Channels

The terminals of the data ports are not displayed, because all terminals of a data port must use one and the same frequency and segment resolution. Therefore, you cannot connect different types of frontends to one data port.

The terminals of a pulse port, however, can be operated at different frequencies.

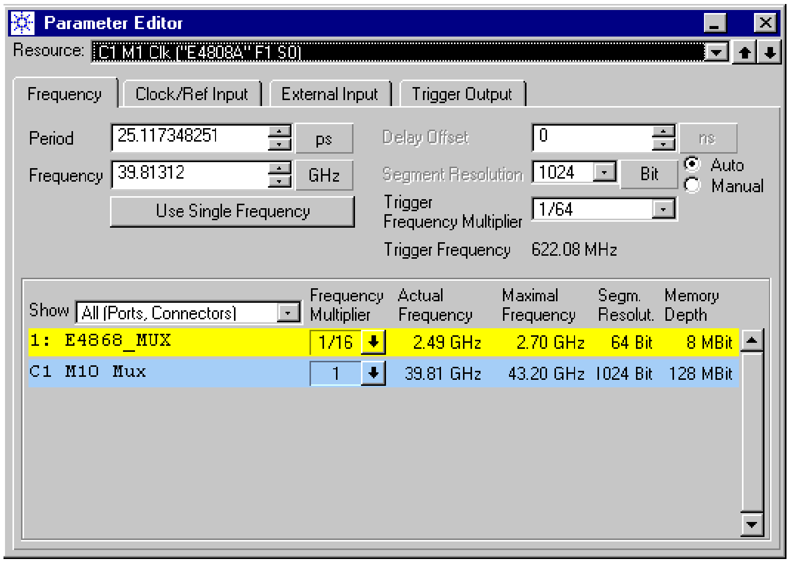
Now you can change the *Frequency Multipliers* of the ports and pulse port terminals.

In the example above, you can also change the *Frequency Multiplier* of the unconnected channel C1 M3 C2. This is possible, because the connector of this channel was enabled (see also *"How to Set Electrical Generator Levels and Termination" on page 232*, *"How to Set Optical Generator Levels" on page 236*, and *"How to Set Analyzer Levels and Termination" on page 245*.

NOTE The *Show* menu allows you to restrict the list to certain items.

ParBERT 43G system

ParBERT 43G systems require multiple frequencies by nature. For a ParBERT 43G system, the Frequency page looks as shown below:



Resource: C1 M1 Clk ("E4808A" F1 S0)

Period 25.117348251 ps

Frequency 39.81312 GHz

Delay Offset 0 ns

Segment Resolution 1024 Bit ● Auto ○ Manual

Trigger Frequency Multiplier 1/64

Trigger Frequency 622.08 MHz

Show All (Ports, Connectors)

	Frequency Multiplier	Actual Frequency	Maximal Frequency	Segm. Resolut.	Memory Depth
1: E4868_MUX	1/16	2.49 GHz	2.70 GHz	64 Bit	8 MBit
C1 M10 Mux	1	39.81 GHz	43.20 GHz	1024 Bit	128 MBit

Figure 92 Frequency Setting for a ParBERT 43G Pattern Generator System

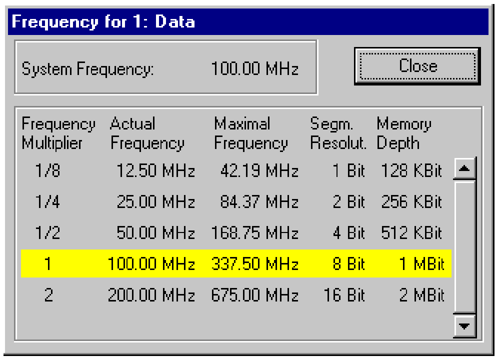
NOTE For a ParBERT 43G system, you may wish to use one of the standard frequencies, such as OC-768, G. 709, G. 975, and so on. Such frequencies can be chosen from the MUX/DEMUX module parameters (see *“How to Set Up a 43G MUX/DEMUX Module” on page 210*).

For a ParBERT 43G system equipped with 3.35 Gbit/s modules and frontends, you may also wish to use its maximum memory capacity and speed. If this is desired and a clock frequency above 42.67 Gbit/s is used, you can increase the global segment resolution to 2048.

If this is done, the analyzers or generators operate at a segment resolution of 128 and have a memory capacity of about 16 Mbit per channel.

3 To change the frequency of a port or channel, click the corresponding down-arrow in the *Frequency Multiplier* column.

You get a list of the available FM factors:



Frequency for 1: Data

System Frequency: 100.00 MHz

Frequency Multiplier	Actual Frequency	Maximal Frequency	Segm. Resolut.	Memory Depth
1/8	12.50 MHz	42.19 MHz	1 Bit	128 KBit
1/4	25.00 MHz	84.37 MHz	2 Bit	256 KBit
1/2	50.00 MHz	168.75 MHz	4 Bit	512 KBit
1	100.00 MHz	337.50 MHz	8 Bit	1 MBit
2	200.00 MHz	675.00 MHz	16 Bit	2 MBit

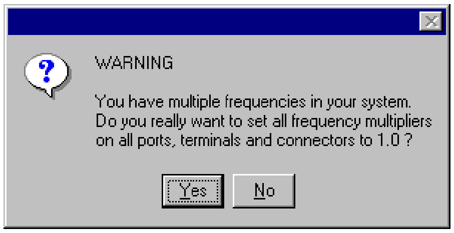
Figure 93 **Individual Frequency Selection**

4 Choose from the list and click *Close*.

In case of a problem, please refer to the tables *“Clock Rates, Segment Resolution, Memory Depth for E4832A Modules” on page 180*, *“Clock Rates, Segment Resolution, Memory Depth for E4861A Modules” on page 180*, and *“Clock Rates, Segment Resolution, Memory Depth for E4810A, E4811A, and E4861B Modules” on page 180*.

Returning to one Single Frequency

If you wish to return to one system frequency for all channels, click *Use Single Frequency* and consider the following warning:



WARNING

You have multiple frequencies in your system.
Do you really want to set all frequency multipliers on all ports, terminals and connectors to 1.0 ?

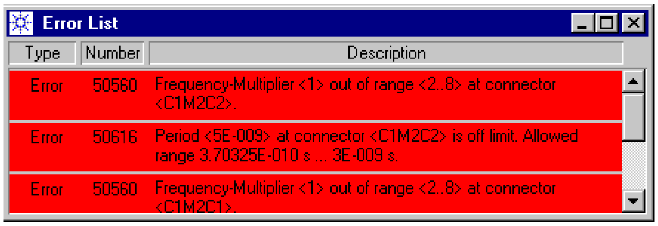
Figure 94 **Return to Single Frequency Setup Warning**

All frequency multipliers will be reset to one. This may conflict with your system configuration.

Troubleshooting

If an error occurs:

1 Click *Show Error(s)* and study the error messages.

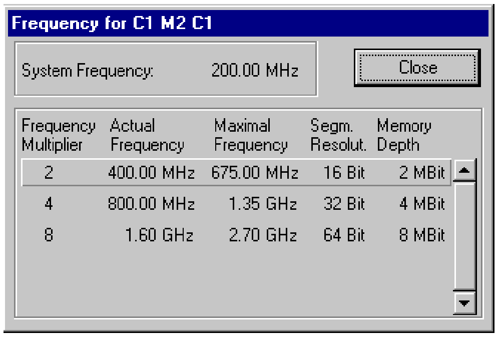


Type	Number	Description
Error	50560	Frequency-Multiplier <1> out of range <2..8> at connector <C1M2C2>.
Error	50616	Period <5E-009> at connector <C1M2C2> is off limit. Allowed range 3.70325E-010 s ... 3E-009 s.
Error	50560	Frequency-Multiplier <1> out of range <2..8> at connector <C1M2C1>.

Figure 95 **Frequency Multiplier Error Examples**

2 Locate the connectors that are causing the problem.

3 Click the down-arrow of these connectors in the *Frequency Multiplier* column. You get a list of the available FM factors:



System Frequency: 200.00 MHz

Frequency Multiplier	Actual Frequency	Maximal Frequency	Segm. Resolut.	Memory Depth
2	400.00 MHz	675.00 MHz	16 Bit	2 MBit
4	800.00 MHz	1.35 GHz	32 Bit	4 MBit
8	1.60 GHz	2.70 GHz	64 Bit	8 MBit

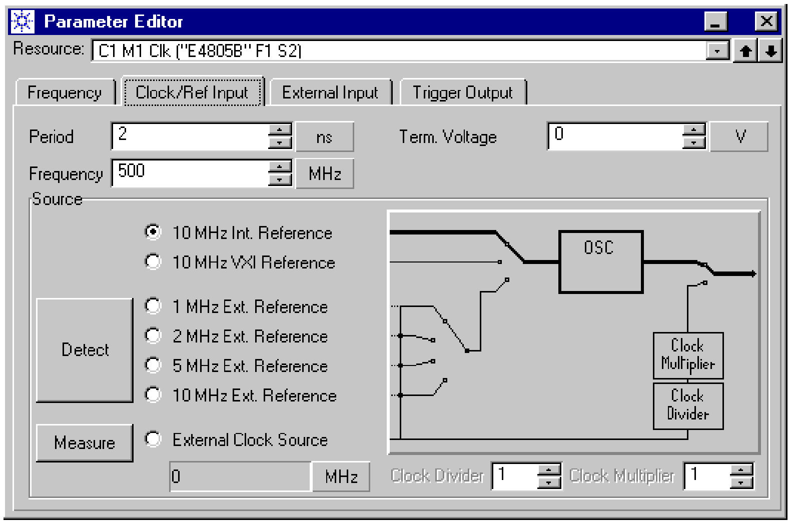
Figure 96 **Frequency Multiplier Range**

4 Choose valid FM factors for the connectors, and the error will disappear.

How to Choose the Clock Source

If you wish to use an external clock, this signal has to be connected to the CLOCK / REF INPUT of the central clock module. The external clock can be used as a reference (an input to the built-in PLL-controlled oscillator) or directly.

1 Open the Parameter Editor for the clock module (see *"How to Start the Parameter Editor for Global Parameters" on page 176*). Click the *Clock/Ref Input* tab.



Parameter Editor

Resource: C1 M1 Clk ("E4805B" F1 S2)

Period 2 ns

Term. Voltage 0 V

Frequency 500 MHz

Source

● 10 MHz Int. Reference
○ 10 MHz VXI Reference
○ 1 MHz Ext. Reference
○ 2 MHz Ext. Reference
○ 5 MHz Ext. Reference
○ 10 MHz Ext. Reference
○ External Clock Source

0 MHz

Clock Divider 1 Clock Multiplier 1

Figure 97 Clock / Reference Input page

2 Enable the type of clock.

Using an External Reference Clock

External reference clocks can have a frequency of 1, 2, 5, or 10 MHz. To use an external clock as a reference to the PLL-oscillator:

1 Connect the clock signal to the CLOCK / REF INPUT .

2 Click the *Detect* button.

The clock is automatically identified and connected.

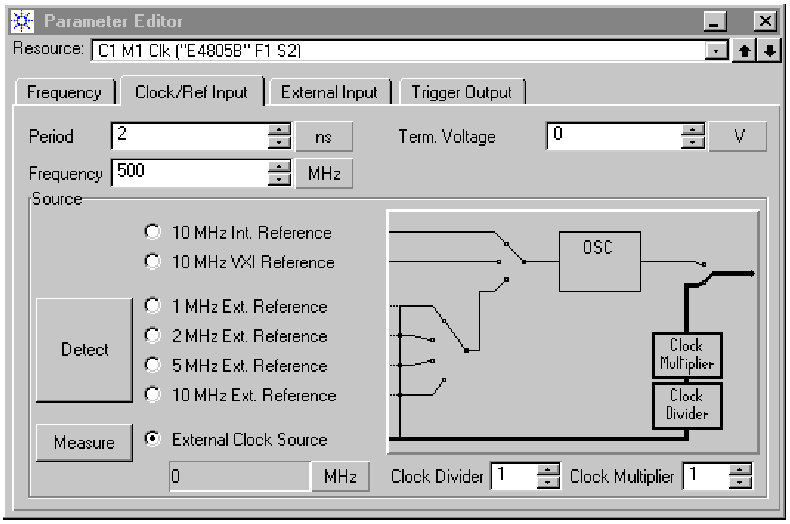
Using an External Source Clock

The frequency of the external source clock must exceed 1.302083 MHz. The maximum frequency is:

- 3.35 GHz for the E4805B clock module
- 10.8 GHz for the E4808A clock module

To use an external clock as a direct source:

1 Connect the clock signal to the CLOCK / REF INPUT .



Resource: C1 M1 Clk ("E4805B" F1 S2)

Period 2 ns

Frequency 500 MHz

Term. Voltage 0 V

Source

○ 10 MHz Int. Reference
○ 10 MHz VXI Reference
○ 1 MHz Ext. Reference
○ 2 MHz Ext. Reference
○ 5 MHz Ext. Reference
○ 10 MHz Ext. Reference
● External Clock Source

0 MHz

Clock Divider 1

Clock Multiplier 1

Figure 98 Using an External Source Clock

2 Click the *Measure* button.

The clock frequency is measured and the clock is automatically connected.

NOTE A built-in phase locked loop will lock onto the external source clock. This may take up to 100 ms. When using an external source clock, observe the PLL lock indicator in the upper right-hand corner of the user interface. It turns red if the PLL could not lock or has lost its synchronization.

3 Decide on using the *Clock Divider* and/or *Clock Multiplier*.

Once you have connected an external source clock, you can use the *Clock Divider* and the *Clock Multiplier*.

The clock multiplier enables you to multiply the applied clock frequency by any integer factor between 1 and 256. The clock divider enables you to divide the applied clock frequency by any integer divisor between 1 and 256.

NOTE If you are using both divider and multiplier, note the following limitation: The product of the multiplying factor and the divisor must not exceed 1024.

For example, if you have set the multiplier to 256, you cannot set the divider to a divisor greater than 4.

Note also that the divider is free-running. If it is set to a divisor other than one, no information about the resulting phase is available outside the system. This has to be considered, if two systems are to be operated synchronously, for example, a stimulating and a separate analyzing system.

Generally, there is no problem, if just a frequency relation is needed, and the systems are synchronized using the automatic synchronization capabilities of the analyzers.

However, phase information may be required, if a second system is to be started synchronously. If this is the case, an external divider has to be used. This divider could be a generator channel of the stimulating system.

For example, for a division by 33, you would assign a pattern consisting of 16 ones and 17 zeros.

TIP If you are using an oscilloscope, this should be triggered by the lowest frequency of the system. Otherwise it will show multiple clock phases.

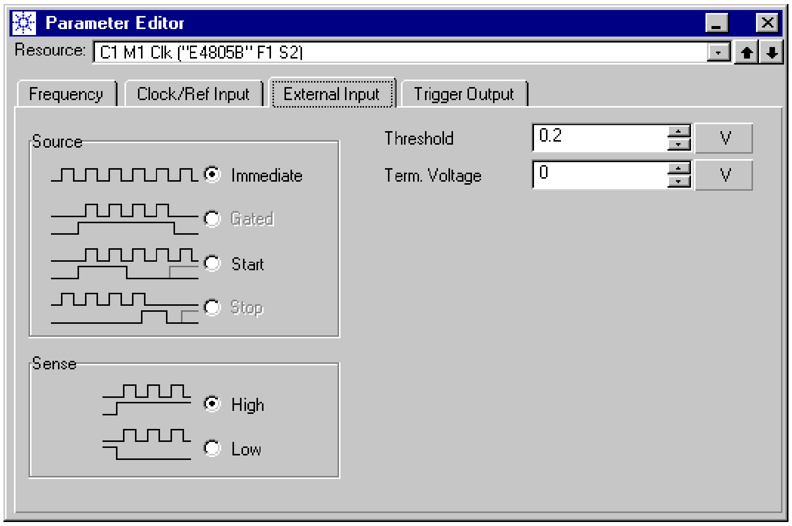
Example: A clock signal is divided by 7 and multiplied by 16 to get a 16:7 clock ratio. The oscilloscope should be triggered by the lowest frequency, which in this case is the clock frequency divided by 7.

How to Set the Characteristics of the External Input

The EXT INPUT connector of the master clock module can be used to start and stop the system.

1 Open the Parameter Editor for the clock module (see *"How to Start the Parameter Editor for Global Parameters" on page 176*). Choose the *External Input* page.

The External Input page shows the alternatives:



Parameter Editor

Resource: C1 M1 Clk ("E4805B" F1 S2)

Frequency | Clock/Ref Input | External Input | Trigger Output

Source
● Immediate
○ Gated
○ Start
○ Stop

Sense
● High
○ Low

Threshold 0.2 V

Term. Voltage 0 V

Figure 99 External Input Configuration Page

2 Choose the desired start/stop mode. Options are:

– *Immediate*:

The timing system is not influenced by the EXTERNAL INPUT. It is started either manually by the user with a mouse click or by a software command.

– *Gated*:

NOTE Gated mode is only supported if the system contains exclusively E4832A modules.

The timing system is armed by pressing the *Run* icon. The user interface displays HALTED as long as the system is not started.

The timing system is started and stopped according to the chosen *Sense* polarity and *Threshold*.

The system is also stopped, if the sequence terminates. The user interface displays RUNNING, HALTED, or STOPPED. See also *"Trigger-Controlled Stop" on page 76*.

- *Start*:

  The timing system is armed by pressing the *Run* icon. The user interface displays HALTED as long as the system is not yet started. The timing system is started with the first edge of the chosen *Sense* polarity that exceeds the *Threshold*. The system is stopped, if the *Stop* icon is pressed or the sequence terminates.

- *Stop*:

NOTE Stop mode is only supported if the system contains exclusively E4832A modules.

The timing system is started by pressing the *Run* icon. The timing system is halted by the first edge of the chosen *Sense* polarity that exceeds the *Threshold*. It is stopped, if the *Stop* icon is pressed or the sequence terminates. See also *"Trigger-Controlled Stop" on page 76*.

3. Specify the trigger *Sense* polarity.
4. Enter an appropriate *threshold* voltage for the external input.
5. Enter an appropriate *termination voltage*, if required.

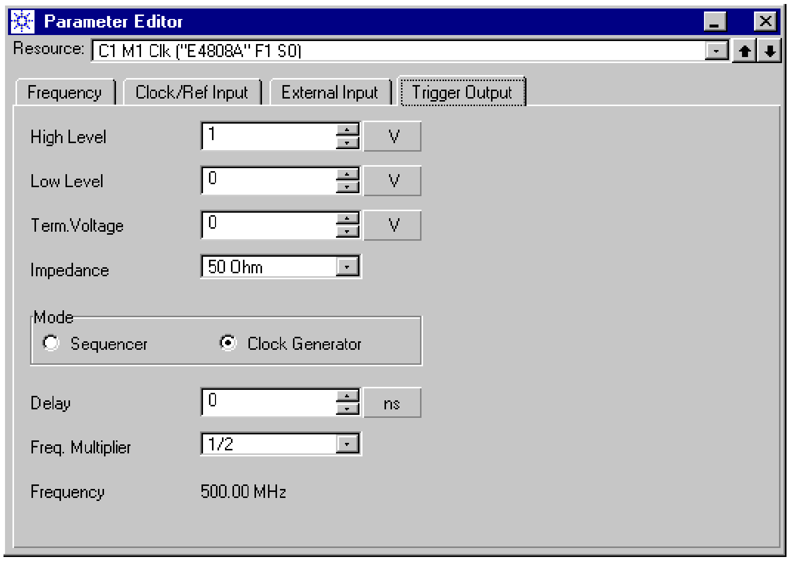
How to Set the Characteristics of the Trigger Output

The TRIGGER OUTPUT connector of the central clock module can generate a clock signal or can be used to inform other instruments upon an event. This event can be:

- Start of a data block within the overall test sequence
- An event detected by the system (see *“Event Handling Principles” on page 104*).

To set the trigger output characteristics:

1 Open the Parameter Editor for the clock module (see *“How to Start the Parameter Editor for Global Parameters” on page 176*). Choose the *Trigger Output* page.



Resource: C1 M1 Clk ("E4808A" F1 S0)

High Level 1 V

Low Level 0 V

Term.Voltage 0 V

Impedance 50 Ohm

Mode: ○ Sequencer ● Clock Generator

Delay 0 ns

Freq. Multiplier 1/2

Frequency 500.00 MHz

Figure 100 Trigger Output Configuration Page

2 Set the voltages and output impedance.

3 Select the Mode.

– *Clock Generator*: generates a clock signal derived from the system clock.

– *Sequencer*: Generates a transition from high to low or vice versa.

4 Set the *delay*, if desired. The delay refers to the system clock.

5 *Frequency Multiplier* is only available in *Clock Generator* mode. This is the *Trigger Frequency Multiplier* of the *Frequency* page.

Below 675 MHz, the trigger clock frequency by default equals the system clock frequency. By setting the *Trigger Frequency Multiplier* to a factor other than 1, you can change the trigger clock frequency.

For example, if you have chosen a system clock frequency of 250 MHz, you can set the trigger clock frequency to 500 MHz.

You can choose one of the factors from the list. The list shows the FMR (see *"How to Set the Clock Frequency" on page 177*).

NOTE The master clock frequency is limited to 675 MHz. If you have set a system clock frequency above 675 MHz, you have to adjust the *Trigger Frequency Multiplier.*

For example, if you have chosen a system clock frequency of 2.7 GHz, the maximum *Trigger Frequency Multiplier* setting is 1/4th.

Connecting the DUT

To connect the device under test (DUT) with the Agilent 81250 system, you can:

- Create signal ports for the DUT.

  There are data input ports, data output ports, and pulse ports. You can create electrical and optical ports.

- Specify the port characteristics.

  You can add terminals to ports. You can also remove terminals from a port, rename the port or delete it.

NOTE Terminals (individual pins) can only be defined within a port.

- Change terminal characteristics.

  You can connect the terminals with suitable module connectors. You can also move a terminal, rename it, or disconnect it.

- Change connector characteristics.

  If your system includes modules that support optical and electrical operation (like the E4810A optical-electrical data generator module), you can switch between optical and electrical mode.

All this can be done with the Connection Editor.

After at least one terminal has been connected, the Connection Editor can also be used to access the Parameter Editor for setting port and terminal or channel specific properties.

NOTE Of course, physical connections are required as well. But you need only take care that the physical connections match the setup shown in the Connection Editor.

ParBERT 43G systems

You may be operating a ParBERT 43G system: These systems, usually delivered as bundles, are preconfigured (see also *“ParBERT 43/45G Systems” on page 107*), but can also be set up individually. Here, the frontends are connected to a MUX or DEMUX module. You can easily access the parameters of the MUX/DEMUX modules from the Connection Editor.

This chapter explains how to use the Connection Editor:

- *"How to Start the Connection Editor" on page 197*
- *"How to Create a Port" on page 198*
- *"How to Change the Characteristics of a Port" on page 201*
- *"How to Change the Characteristics of a Terminal" on page 203*
- *"How to Set the Mode of an Optical-Electrical Connector" on page 209*
- *"How to Set Up a 43G MUX/DEMUX Module" on page 210*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo

♦ In the *Tutorial*, select *"Creating a Graphical Image of Your Test Setup"*.

How to Start the Connection Editor

The Connection Editor is the first window that comes up automatically after starting the user interface.

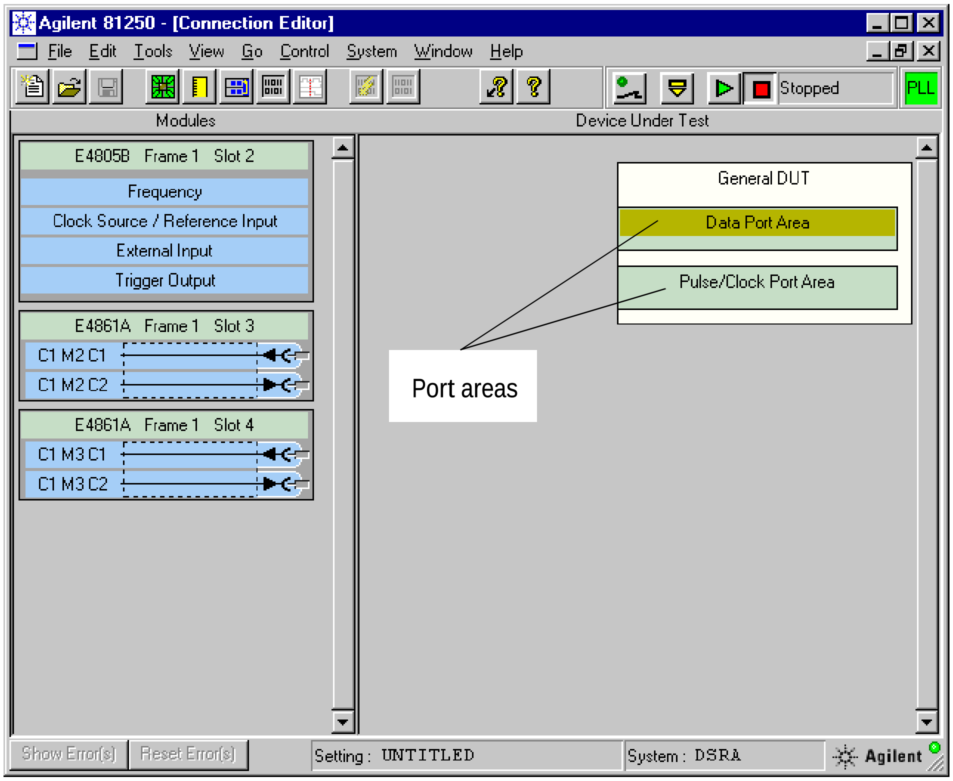
To start the Connection Editor if you have closed its window:

1 Click the Connection Editor icon in the tool bar.

Alternatively, you can also start the Connection Editor from the *Go* menu.

Contents of the Connection Editor Window

At the left-hand side is the *Modules* section. Here, the Connection Editor shows an image of the instrument, including all its modules, channels, and connectors. The following figure shows an example.



Setting : UNTITLED

System : DSRA

Figure 101 Connection Editor Window

The channel identifiers consist of: Cx My Cz (ClockgroupNumber – ModuleNumber – ConnectorNumber), as for instance C1 M3 C2.

The connectors are also marked with arrows. So you know at a glance whether a connector belongs to an output channel (data generator) or an input channel (data analyzer).

At the right-hand side, there is a template for setting up an image of the DUT. You can add:

- Data input ports
- Data output ports
- Pulse ports
- Port terminals representing the individual pins of the port

NOTE If you are operating a ParBERT 43G system, this template is preconfigured.

For these systems, the “DUT” that receives generated data is the E4868A multiplexer module. The “DUT” that returns the data to be analyzed is the E4869A demultiplexer module. The “DUT” has one port with 16 terminals. All the terminals are readily connected to generator or analyzer frontends, respectively.

For basic information on ParBERT 43G systems see *“ParBERT 43/45G Systems” on page 107*. For examples of the ParBERT 43G Connection Editor window see *“ParBERT 43G Software Support” on page 112*.

How to Create a Port

If you start from scratch, the DUT area of the Connection Editor shows only an empty template. You have to define ports (usually i/o buses, but also ports for clocks, latches, strobes, etc.) and single terminals (DUT pins).

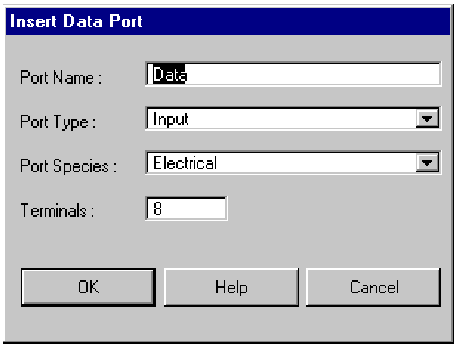
DUT data ports receive or return data streams. DUT pulse ports receive clocks and pulses from the Agilent 81250 system.

To create a port:

1 Click on the green port area (either Data Port Area or Pulse Port Area) with the right mouse button to open the context menu.

2 For data ports you can choose the *Port Type*. Select either *Input Port* or *Output Port*. Pulse ports are always DUT input ports.

   The *Insert Data* or *Insert Pulse Port* dialog box is displayed.



Insert Data Port

Port Name : Data

Port Type : Input

Port Species : Electrical

Terminals : 8

Figure 102 Insert Data Port Dialog

If you have chosen to insert a data port, you can still switch between DUT input and output port.

NOTE The terminals of an input port can only be connected to data generator channels, the terminals of an output port only to analyzer channels.

3 Enter a suitable *Port Name*. By default, the terminals get the port name and are automatically numbered.

4 Choose the *Port Species*: electrical or optical.

NOTE The terminals of an electrical port can only be connected to electrical connectors, the terminals of an optical port only to optical connectors.

5 Specify the number of terminals.

6 Click *OK*.

The Connection Editor shows the new port.

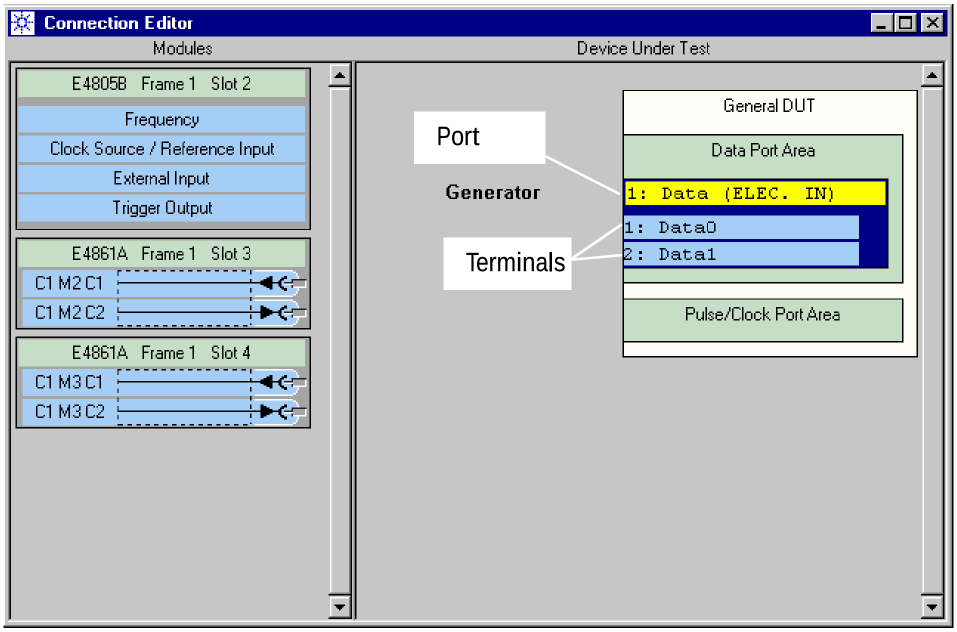


Figure 103 Display of Ports and Terminals

The terminals can be renamed or rearranged if required. See *"How to Rename a Terminal" on page 203* and *"How to Move a Terminal" on page 204*.

How to Change the Characteristics of a Port

Although all actions can also be selected from the menus in the main window, it is a lot more convenient to employ the individual context menu of each item.

To open the context menu of a certain port, place the cursor on the port’s header and click the right mouse button.

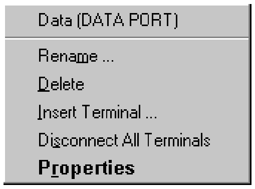


Figure 104 Context Menu of a Port

The available options are:

- Delete the port
- Rename the port
- Add terminals to the port

After one or several of the port terminals have been connected to instrument connectors, you have the additional options to:

- Disconnect all terminals
- Set or change the port properties

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial*.

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo

How to Delete a Port

To delete a port:

1 Open the port’s context menu by clicking the port header with the right mouse button.

2 Choose *Delete*.

3 Confirm.

How to Rename a Port

To rename a port:

1 Open the port’s context menu by clicking the port header with the right mouse button.

2 Choose *Rename*.

3 Specify a new name for the port.

4 Confirm with Enter or click *OK*.

How to Add a Terminal to a Port

To add a terminal to a port:

1 Open the port’s context menu by clicking the port header with the right mouse button.

2 Choose *Insert Terminal*.

The window shows the default terminal name (port name plus number) and suggests the position at the end of the terminal list.

3 Specify the desired terminal name.

4 Accept the suggested position or enter one of the occupied positions to insert the new terminal there.

5 Confirm with Enter or click *OK*.

NOTE The terminal’s name and position can be changed at any time. Although permitted, you should not use one name for several terminals of a port in order to avoid confusion.

How to Change the Characteristics of a Terminal

Although all actions can also be selected from the menus in the main window, it is a lot more convenient to employ the individual context menu of each item.

To open the context menu of a terminal, place the cursor on it and click the right mouse button.

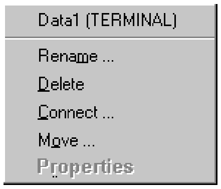


Figure 105 **Context Menu of a Terminal**

The available options are to:

- Rename the terminal
- Delete the terminal
- Connect the terminal to the instrument
- Move the terminal

If a terminal has been connected to an instrument connector, you can also

- Disconnect the terminal
- Set or change the channel properties

How to Rename a Terminal

To rename a terminal:

1. Open the context menu by clicking the terminal with the right mouse button.
2. Choose *Rename*.
3. Enter a new name for the terminal.
4. Confirm with Enter or click *OK*.

How to Delete a Terminal

To delete a terminal:

1 Open the terminal’s context menu by clicking it with the right mouse button.

2 Choose *Delete*.

3 Confirm.

How to Move a Terminal

To move a terminal within its port:

1 Open the terminal’s context menu by clicking it with the right mouse button.

2 Choose *Move*.

3 Overwrite the present location by the desired location.

If the terminal is placed at the last position of the port, you can only move it upward.

4 Confirm with Enter or click *OK*.

Shortcut: Click the terminal with the left mouse button and drag it to the desired position.

How to Connect a Terminal

There are several ways to connect the terminals of the DUT to the connectors of the instrument frontends:

- *“Connecting a Terminal With the Keyboard” on page 206*
- *“Connecting a Terminal With the Mouse” on page 207*
- *“Inserting and Connecting a Single Terminal With the Mouse” on page 207*
- *“Inserting and Connecting Multiple Terminals With the Mouse” on page 208*

NOTE The terminals of a data input port or a pulse port can only be connected to generator channels. The terminals of a data output port can only be connected to analyzer channels.

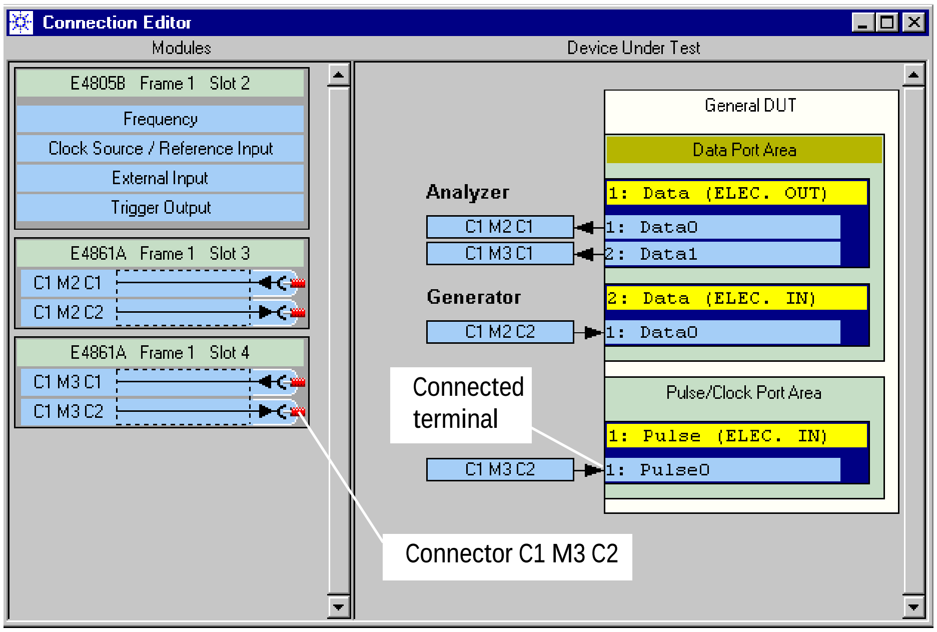


Figure 106 Display of Terminal Connections

The figure above shows electrical ports and connections. Optical connections are displayed as depicted in the following figure.

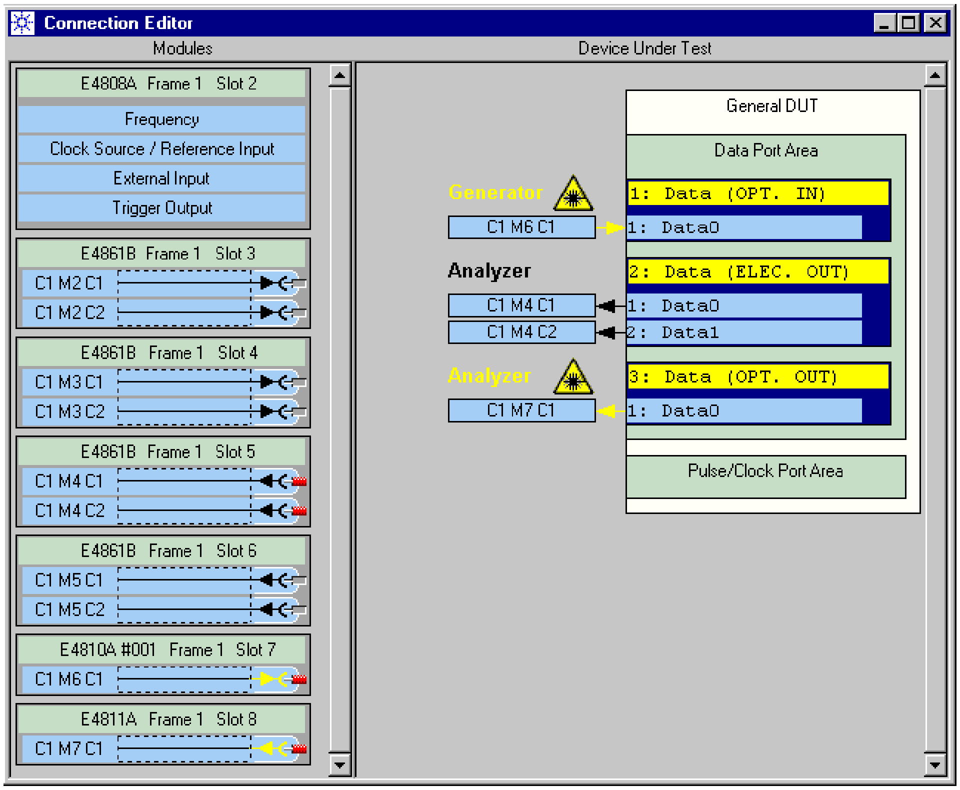


Figure 107 Display of Optical Terminal Connections

Optical connections appear yellow and are marked with the laser symbol.

Connecting a Terminal With the Keyboard

To connect a terminal with the keyboard:

1. Click the right mouse button to open the terminal’s context menu.
2. Choose *Connect*.
3. Enter the identifier of the desired channel: ClockgroupNumber, ModuleNumber, and ConnectorNumber.

   Remember that electrical terminals can only be connected to electrical connectors, optical only to optical.
4. Confirm with Enter or click *OK*.

Connecting a Terminal With the Mouse

To connect a terminal with the mouse:

1. Position the cursor on the terminal.
2. Press the left mouse button. Hold the mouse button depressed and drag the cursor (which changes its shape) onto an empty connector of appropriate type.

   Remember that electrical terminals can only be connected to electrical connectors, optical only to optical.
3. Release the mouse button.

Inserting and Connecting a Single Terminal With the Mouse

You can create a new one-terminal port or insert a new terminal into an existing port and connect it in one go:

1. Position the cursor on the frontend connector in the *Modules* section.
2. Press the left mouse button. Hold the mouse button depressed and drag the cursor (which changes its shape) onto the DUT.
   - If you place the cursor on the green header of the *Data Port Area*, a new port with a single terminal will be created. The *Port Species* (electrical or optical) depends on the connector type.
   - If you place the cursor on a terminal, this terminal will be connected to the module channel.

     Remember that electrical connectors can only be connected to electrical terminals, optical only to optical.
   - If you place the cursor on the border line of a terminal (the border turns red), a new terminal will be inserted at this point.
3. Release the mouse button to establish the connection.

Inserting and Connecting Multiple Terminals With the Mouse

You can create new ports with multiple terminals or add multiple terminals to an existing port and connect them in one go:

1 In the *Modules* section place the cursor on a module label to connect all its connectors at once.

2 Press the left mouse button. Hold the mouse button depressed and drag the cursor (which changes its shape) onto the DUT.

 – If you place the cursor on the green header of the *Data Port Area*, one or two new port(s)—an input or output port or both—will be created with all (matching) channel connectors being connected to terminals of this new port.

 – If you place the cursor on an existing terminal, this terminal and the following ones will be connected to the module connectors. No new terminals will be created.

 Remember that electrical connectors can only be connected to electrical terminals, optical only to optical.

 – If you place the cursor on the upper line of an existing terminal (the border turns red), new terminals will be inserted at this location. They will be connected to all matching connectors of the selected module.

3 Release the mouse button to establish the connections.

How to Disconnect a Terminal

To disconnect a terminal:

1 Open the terminal’s context menu by clicking the terminal with the right mouse button.

2 Choose *Disconnect*.

3 Confirm.

To disconnect all terminals of a port, choose *Disconnect* from the port’s context menu.

How to Set the Mode of an Optical-Electrical Connector

If your system includes modules that support both optical and electrical operation (like the E4810A optical-electrical data generator module), you can switch between optical and electrical mode.

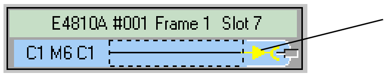
To change the mode:

1 Ensure that the connector is unconnected.

2 Double click the connector

or

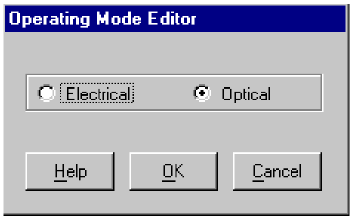
right-click the connector and choose *Properties*.



Click here

Figure 108 Optical Connector

This opens the Operating Mode Editor.



○ Electrical ● Optical

Figure 109 Operating Mode Editor

3 Set the mode and click *Ok*.

When an optical-electrical generator is in electrical mode, the laser is switched off.

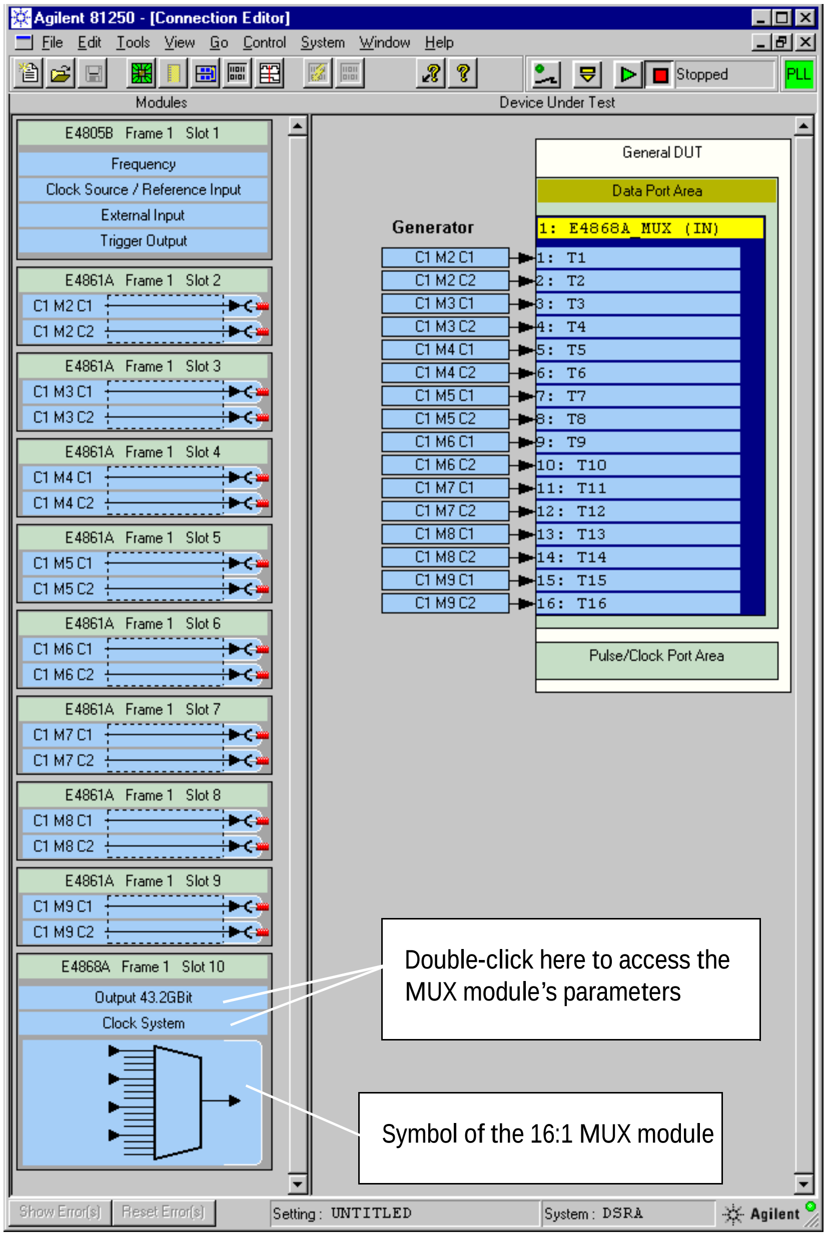
In electrical mode, optical-electrical generators or analyzers can be used and set up like standard electrical generators or analyzers.

For example, an E4810A data generator in electrical mode behaves like an E4862B 3.35 Mbit/s frontend.

In optical mode, different signal parameters have to be specified (for details see *"How to Set Up a DUT Input Port or Generator Channel" on page 226* and *"How to Set Up a DUT Output Port or Analyzer Channel" on page 244*).

How to Set Up a 43G MUX / DEMUX Module

The simplest way to set up a multiplexer or demultiplexer module of a ParBERT 43G system is via the Connection Editor. For a ParBERT 43G pattern generator system, the Connection Editor appears as shown:



Double-click here to access the MUX module's parameters

Symbol of the 16:1 MUX module

Figure 110 Connection Editor of a ParBERT 43G Pattern Generator System

A 43.2 Gbit/s MUX or DEMUX module has two types of parameters:

- MUX module: *Output 43 GBit* and *Clock System* (as shown in the figure above)
- DEMUX module: *Input 43 GBit* and *Clock System*

Double-click one of the parameter types to check or change the module parameters. This opens the Parameter Editor for the module. The Parameter Editor has two pages, and you can toggle between the two parameter types.

NOTE A MUX or DEMUX module can hardly be compared with a clock or a data generator/analyzer module, although it combines some features of both basic categories.

You should consider a ParBERT 43G system as a self-contained unit for testing very high speed MUX/DEMUX devices. In this context, setting channel parameters to individual values is useless and, worse, bears the risk of damaging the MUX or DEMUX module.

Therefore, most of the usual port and channel parameters are preset to appropriate values and cannot be changed. Some auxiliary functions are also disabled. See *"ParBERT 43G Software Support" on page 112*.

As long a you have not more than the required number of generator/analyzer modules and frontends installed, the two buttons shown in the Connection Editor (*Output/Input 43 GBit* and *Clock System*) provide easy access to all relevant parameters.

The following table summarizes the parameters of the MUX/DEMUX modules and provides a comparison of the “A” and “B” modules.

Table 16 Parameters of MUX/DEMUX Modules

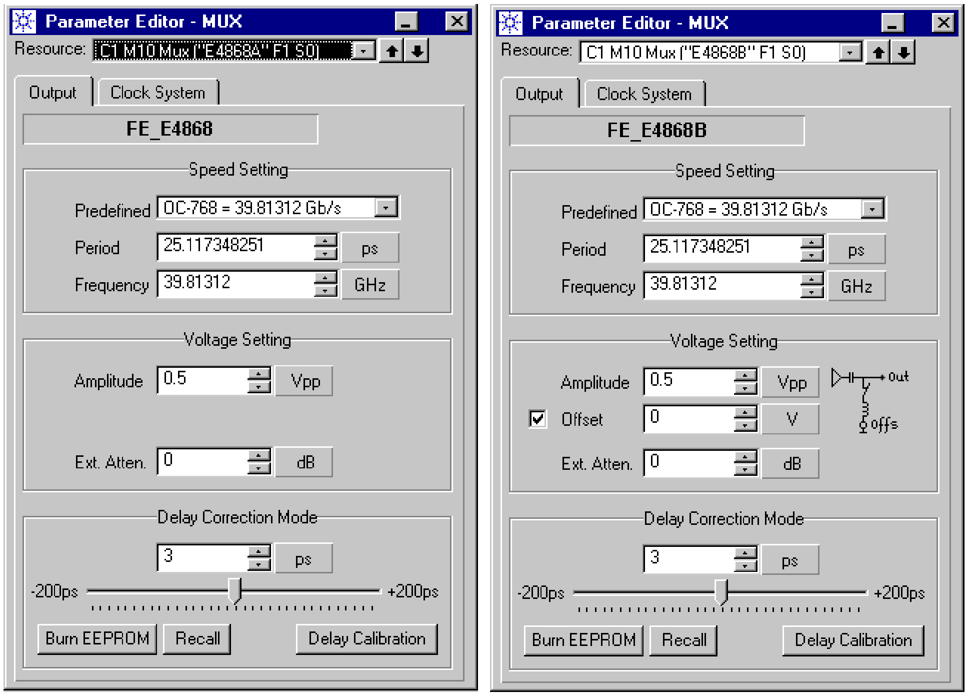
Parameter Group	Parameter and Command	E4868A (MUX)	E4868B (MUX)	E4869A (DEMUX)	E4869B (DEMUX)
Speed Parameters	Predefined Frequency	+	+	+	+
	Period	+	+	+	+
	Frequency	+	+	+	+
Voltage Configuration	Amplitude	+	+	–	–
	Offset	–	+	–	–
	Offset Enabling	–	+	–	–
	External Attenuator	+	+	–	–
Delay Correction/ Calibration	Delay Value	+	+	–	–
	Delay Mode	+	+	–	–
	Delay Storage Handling	+	+	–	–
Offset Compensation	Offset Comp. Vernier	–	–	+	–
Manual Threshold	Threshold Vernier	–	–	–	+
Manual Sampling Phase Adjustment	Sampling Phase Vernier	–	–	–	+
Clock System	External Clock Input	+	+	+	+
	Clock Output (Subrate)	+	+	–	–

For detailed instructions see:

- *“How to Change the Output Parameters of a MUX Module” on page 213*
- *“How to Change the Clock Routing of a MUX Module” on page 218*
- *“How to Change the Input Parameters of a DEMUX Module” on page 219*
- *“How to Change the Clock Routing of a DEMUX Module” on page 222*

How to Change the Output Parameters of a MUX Module

The E4868A and E4868B MUX modules have slightly different output parameters:



Parameter Editor - MUX

Resource: C1 M10 Mux ("E4868A" F1 S0)

FE_E4868

Speed Setting

Predefined: OC-768 = 39.81312 Gb/s

Period: 25.117348251 ps

Frequency: 39.81312 GHz

Voltage Setting

Amplitude: 0.5 Vpp

Ext. Atten.: 0 dB

Delay Correction Mode: 3 ps

Parameter Editor - MUX

Resource: C1 M10 Mux ("E4868B" F1 S0)

FE_E4868B

Speed Setting

Predefined: OC-768 = 39.81312 Gb/s

Period: 25.117348251 ps

Frequency: 39.81312 GHz

Voltage Setting

Amplitude: 0.5 Vpp

☑ Offset: 0 V

Ext. Atten.: 0 dB

Delay Correction Mode: 3 ps

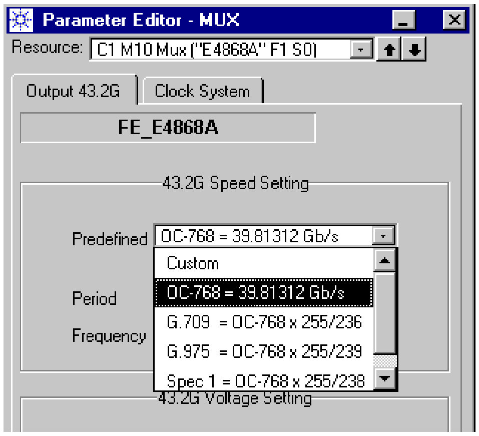
Figure 111 E4868A and E4868B MUX Module Output Parameters

Speed Setting

1 Set the output *Speed*.

The displayed frequency and period are the same as in the Frequency page of the clock module. The *Period* is always the reciprocal of the *Frequency*.

Standard frequencies can be chosen from the list of predefined data rates.



Parameter Editor - MUX

Resource: C1 M10 Mux ("E4868A" F1 S0)

Output 43.2G | Clock System

FE_E4868A

43.2G Speed Setting

Predefined: OC-768 = 39.81312 Gb/s

Figure 112 Predefined Frequencies

NOTE For a ParBERT 43G system equipped with 3.35 Gbit/s modules and frontends, you may wish to use its maximum memory capacity and speed. If this is desired and a clock frequency above 42.67 Gbit/s is used, you can increase the global segment resolution from 1024 to 2048 from the Frequency page of the clock module.

If this is done, the generators operate at a segment resolution of 128 bits and have a memory capacity of about 16 Mbit per channel.

See also *"How to Use Multiple Frequencies" on page 182*.

Voltage Setting

2 Set the desired *Amplitude*. This is the signal voltage to be applied to the DUT.

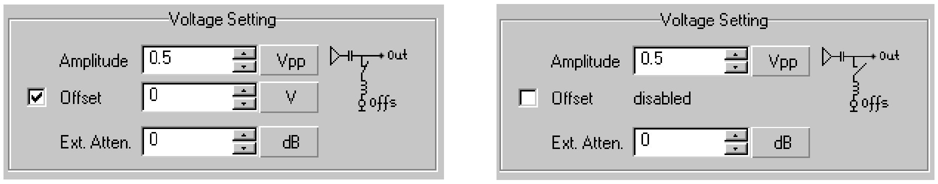
The full amplitude voltage range without attenuator is 0.5 V to 2.5 V.

TIP If you are using an external attenuator, you should first enter its setting and then specify the signal amplitude and the offset. This helps to avoid error messages.

3 If you are using an E4868B MUX module, you can add a DC offset to the generated signal.

By default, the offset is enabled. Without attenuator, you can specify a voltage between zero and one Volt.

You can also disable the offset, as shown in the figure below.



Voltage Setting

Amplitude 0.5 Vpp

☑ Offset 0 V

Ext. Atten. 0 dB

Voltage Setting

Amplitude 0.5 Vpp

□ Offset disabled

Ext. Atten. 0 dB

Figure 113 Offset Setting

When the offset is disabled, the module generates a pure AC coupled signal. When you enable the offset again, your last value is restored.

4 If you use an external attenuator, enter the *Ext. Attenuator* setting.

An external attenuator can be used to reduce the signal amplitude without increasing the signal-to-noise ratio.

The following figure illustrates its connection. Note that the offset generator is only available for the E4868B MUX module.

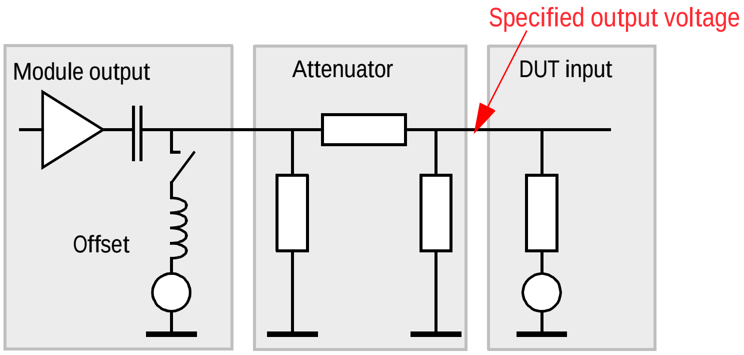


Figure 114 Using an External Attenuator

If you have specified an external attenuator, the range of the amplitude voltage is recalculated. The software considers the attenuator as a part of the MUX module which reduces the amplitude range.

For example, if you have set the attenuator to 6 dB, the desired signal amplitude at the DUT has to be within 0.25 V to 1.25 V. If you had set the attenuator to 20 dB, the legal amplitude voltage range would be 1/10th of the full amplitude, which means 0.05 V to 0.25 V.

The same calculation applies to the offset voltage. For example, if you have set the attenuator to 6 dB, the offset must remain below 0.5 V.

Note that the recalculation assumes that a 50 Ω termination is used (which is mandatory).

Delay Correction and Delay Calibration

Delay Correction and *Delay Calibration* are optional functions. You can click the corresponding button at the lower right-hand side of the page to toggle between these options.

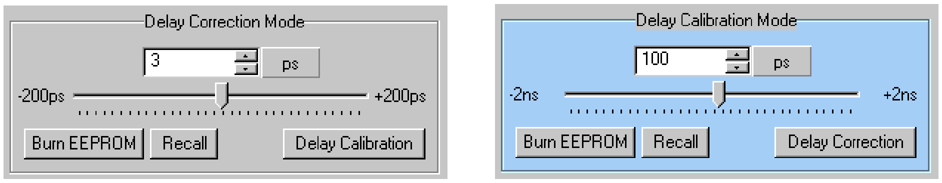


Figure 115 Delay Correction and Calibration Modes

If you receive one of the ParBERT 43G bundles, all frontends and MUX/DEMUX modules are factory-calibrated and ready to use.

These two options allow you to add a common delay to all the generator frontends connected to the MUX module.

Delay Correction High-frequency applications are generally sensitive to changes of the environmental conditions. *Delay Correction* allows you to compensate for small changes that, for example, may be caused by operating temperatures that differ from the manufacturing temperature.

Delay Calibration *Delay Calibration* is required if the clock module or the cable set of the MUX module has been exchanged. This mode offers a wider timing range.

Delay Calibration sets the base value for the Delay Correction. If the Delay Calibration value is changed, the Delay Correction value is automatically reset to zero.

The system stores two Delay Calibration values—one for an external clock source, one for the internal clock source. The value that is displayed depends on the currently selected clock source. If you are not sure, you should take a look at the Clock/Ref Input page of the master clock module.

If you change the delay values, your changes take immediate effect and remain effective until the system is switched off. As long as the system is not powered down, your changes remain effective, even if you terminate and restart the ParBERT User Software.

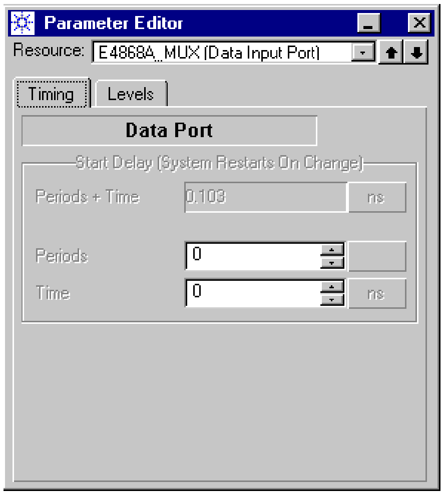
Burn EEPROM button If you wish to save the new values, click the Burn EEPROM button.

This is necessary because the delay values of a MUX module are stored in the module itself. They are independent of the loaded setting.

For details of the delay calibration procedure refer to *"How Do I Calibrate a 43G Pattern Generator?" on page 413*.

Recall button If you have changed the delay values, you can always return to the last saved values by clicking the Recall button.

TIP The actual start delay of the MUX input port is displayed on the Timing page of the port (in the Connection Editor, double-click the yellow port area). In the figure below, the present start delay is 103 ns.



Resource: E4868A_MUX (Data Input Port)

Start Delay (System Restarts On Change)

Periods + Time 0.103 ns

Periods 0

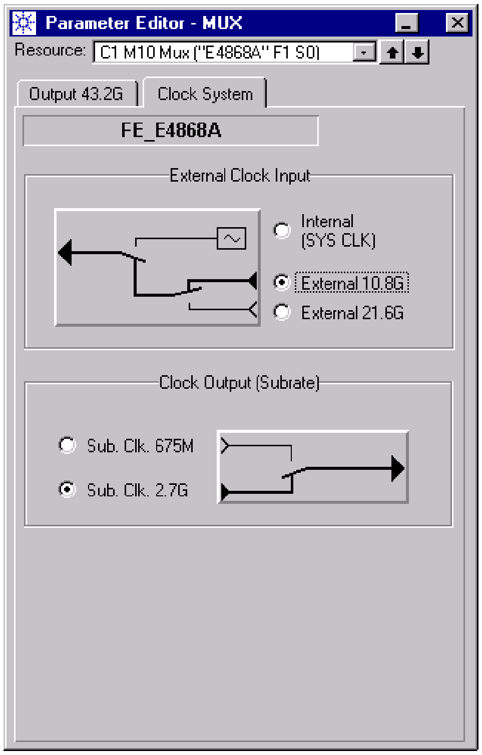
Time 0 ns

Figure 116 Timing Properties of a MUX Input Port

How to Change the Clock Routing of a MUX Module

The Clock routing of a MUX Module can be changed on the Clock System page of the MUX module parameters.

The window clearly shows the capabilities and the present routing.



Resource: C1 M10 Mux ("E4868A" F1 S0)

External Clock Input

○ Internal (SYS CLK)

● External 10.8G

○ External 21.6G

Clock Output (Subrate)

○ Sub. Clk. 675M

● Sub. Clk. 2.7G

Figure 117 **MUX Module Clock Routing**

1 Set the clock input. Choices are:

   – *Internal (SYS CLK)*

     The module uses internally a clock derived from the SYSTEM CLOCK OUTPUT signal provided by a central clock module. The SYS CLK output from the central clock module needs to be connected with the SYS CLK input of the MUX module.

   – *External 10.8 G*

     To obtain more precision, the module uses an external signal in the range of 10 GHz provided by an external clock generator to its Ext. Clk. input connector.

   – *External 21.6 G*

     The module uses an external signal in the range of 20 GHz.

2 Set the subrate clock output.

The subrate clock is derived from the multiplexer clock. The subrate clock is provided for general use at the subrate clock output connector of the MUX module. An internal clock divider allows to choose between two different speeds:

- Sub. Clk. 675M

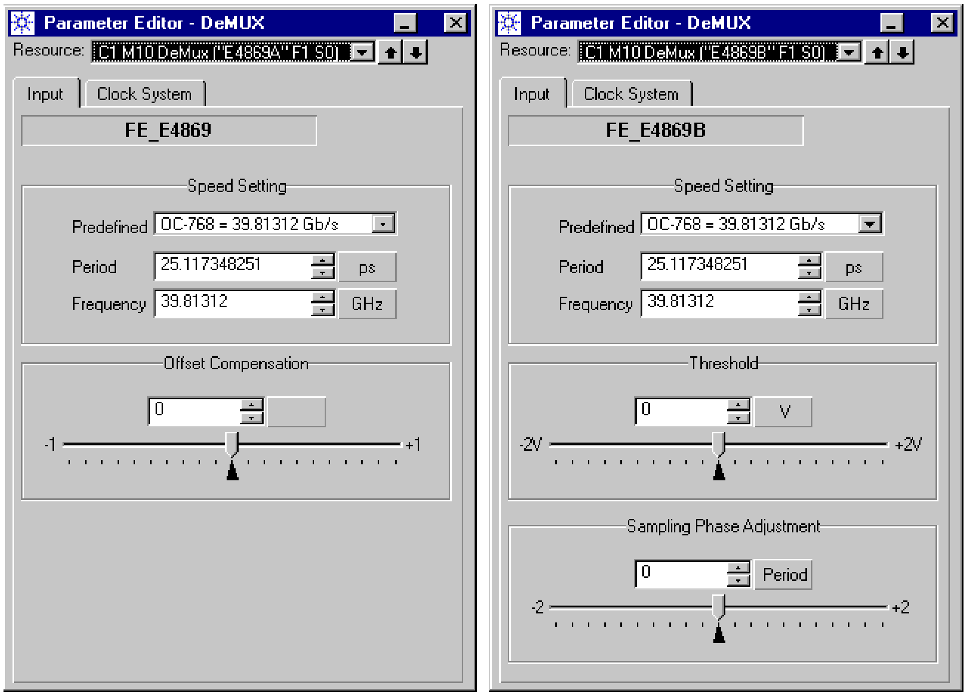
  A signal of up to 675 MHz is provided at the subrate clock output connector.

- Sub. Clk. 2.7G

  A signal of up to 2.7 GHz is provided at the subrate clock output connector.

How to Change the Input Parameters of a DEMUX Module

The E4869A and E4869B DEMUX modules have different input parameters:



Resource: C1 M10 DeMux ("E4869A" F1 S0)

FE_E4869

Predefined: OC-768 = 39.81312 Gb/s

Period: 25.117348251 ps

Frequency: 39.81312 GHz

Offset Compensation: 0

Resource: C1 M10 DeMux ("E4869B" F1 S0)

FE_E4869B

Predefined: OC-768 = 39.81312 Gb/s

Period: 25.117348251 ps

Frequency: 39.81312 GHz

Threshold: 0 V

Sampling Phase Adjustment: 0 Period

Figure 118 E4869A and E4869B DEMUX Module Input Parameters

Speed Setting

♦ Set the expected data rate.

The displayed frequency and period are the same as in the Frequency page of the master clock module. The *Period* is always the reciprocal of the *Frequency*.

Standard frequencies can be chosen from the list of predefined data rates.

NOTE For a ParBERT 43G system equipped with 3.35 Gbit/s modules and frontends, you may wish to use its maximum memory capacity and speed. If this is desired and a clock frequency above 42.67 Gbit/s is used, you can increase the global segment resolution from 1024 to 2048 from the Frequency page of the clock module.

If this is done, the analyzers operate at a segment resolution of 128 bits and have a memory capacity of about 16 Mbit per channel.

See also *"How to Use Multiple Frequencies" on page 182*.

If you are using an E4869B DEMUX module, please continue with *"Threshold and Sampling Phase Adjustment" on page 221*.

Offset Compensation

♦ Decide on adjusting the Offset.

The *Offset Compensation* slider can be used while a test (a BER measurement) is running.

By default, the decision threshold for sampling the incoming signal is set according to the mean value of the data signal. This is adequate for pure PRBS data. In cases where the received pattern has an unbalanced ratio of zeros to ones, the Offset Compensation slider allows you to fine-tune the threshold to an appropriate value.

An inadequate offset can be suspected, if:

- None of the analyzer terminals could synchronize
- All analyzer terminals did synchronize, but exhibit an abnormal high BER
- Some analyzer terminals did synchronize but show a BER close to the synchronization threshold, while others did not synchronize at all

Channels that could not synchronize exhibit no BER test results (the BER counter is disabled, see also *"Automatic Bit Synchronization" on page 92*).

If none of the analyzer channels could synchronize, the BER measurement should be stopped. Now you can move the slider in coarse steps and repeat the test until one or more channels synchronize. The Offset Compensation slider covers a range of about ±100 mV.

As soon as at least one terminal has synchronized, it is recommended to move the slider in small steps and in both directions while the BER test is running. Wait a few seconds after each step and observe the actual bit error rate. Once you have found the slider position of minimum BER, stop and restart the test to ensure that all channels are synchronized.

Threshold and Sampling Phase Adjustment

1 If necessary, use the *Threshold* vernier to adjust the input decision threshold of the E4869B DEMUX module.

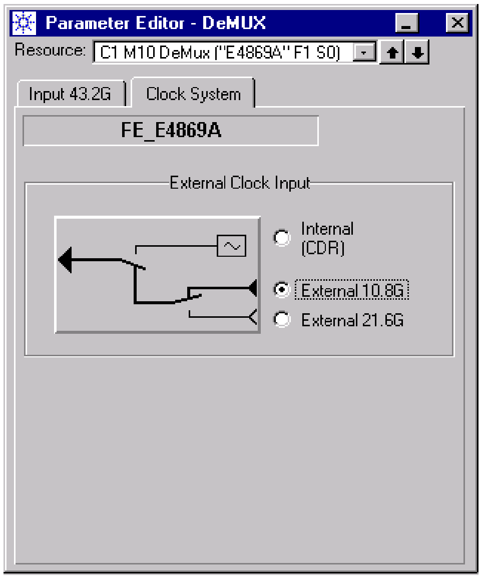
2 If necessary, use the *Sampling Phase* vernier to adjust the sampling point of the E4869B DEMUX module.

This is only required, if you use the system for capturing data. For *Compare and Capture* or *BER* measurements, leave the vernier in zero position and activate *Auto Bit Sync*.

How to Change the Clock Routing of a DEMUX Module

The clock routing of a DEMUX Module can be changed on the Clock System page of the DEMUX module’s parameters.

The window clearly shows the capabilities and the present routing:



Parameter Editor - DeMUX

Resource: C1 M10 DeMux ("E4869A" F1 S0)

Input 43.2G | Clock System

FE_E4869A

External Clock Input

○ Internal (CDR)

● External 10.8G

○ External 21.6G

Figure 119 DEMUX Module Clock Routing

♦ Set the clock source. Choices are:

– *Internal (CDR)*

The module uses a clock signal derived from the incoming serial bit stream. The clock is recovered by the built-in Clock Data Recovery circuitry (CDR).

– *External 10.8 G*

The DUT itself or an external clock generator may provide a clock signal in the range of 10 GHz which represents the timing of the serial bit stream. This signal can be attached to the external clock input connector of the DEMUX module.

– *External 21.6 G*

The DUT or an external clock generator provides a signal in the range of 20 GHz.

Setting Up Ports and Channels

After one or several terminals have been connected to the Agilent 81250 system, you can set up and modify port and channel properties, such as delays, signal formats, voltages, impedances, and so on.

Setting port parameters is an easy way to use the same settings for all connected terminals of that port.

NOTE Before setting port parameters, **all** terminals of that port should be connected with the instrument, because later connected terminals do not automatically get the same parameter settings.

Alternatively, you can set individual parameters for individual channels. Individual channel parameters override port parameters.

To set and change port and channel parameters, the Parameter Editor is used.

NOTE In addition to setting parameters, there is another function that determines channel properties: Two or four 675 Mbit/s data generator channels can be added to produce a combined signal. This is done with the Channel Configuration Editor which can be invoked from the Connection Editor.

This chapter explains how to set up the parameters for ports and channels:

- *"How to Start the Parameter Editor for Ports/Channels" on page 224*
- *"How to Set Up a DUT Input Port or Generator Channel" on page 226*
- *"How to Set Up a DUT Output Port or Analyzer Channel" on page 244*
- *"How to Combine Generator Channels" on page 251*
- *"How to Set Up N4868A Booster Channels" on page 255*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *"Comfortable Control of Signal Parameters"*.

How to Start the Parameter Editor for Ports / Channels

The Parameter Editor has several modes of operation. It distinguishes between global system parameters and parameters for generator and analyzer channels.

The setup of global parameters is described in *"Setting Global System Parameters" on page 175*. The present chapter deals with port and channel parameters.

The Parameter Editor can be started conveniently from the Connection Editor.

To open the Parameter Editor from the Connection Editor, you have the following options:

- Use the left mouse button and double-click a port header or a connected terminal in the *Device Under Test* section, or a channel identifier (like C1 M2 C2) in the *Modules* section.

  The Parameter Editor opens and displays the properties of the selected item.

- Use the right mouse button and click on a port header or a connected terminal in the *Device Under Test* section, or a channel identifier (like C1 M2 C2) in the *Modules* section. Then select *Properties* from the context menu.

  The Parameter Editor opens and displays the properties of the selected item.

NOTE From the *Device Under Test* section of the Connection Editor, you can directly access the parameters of ports and connected terminals.

From the *Modules* section of the Connection Editor, you can directly access the parameters of connected and unconnected channels.

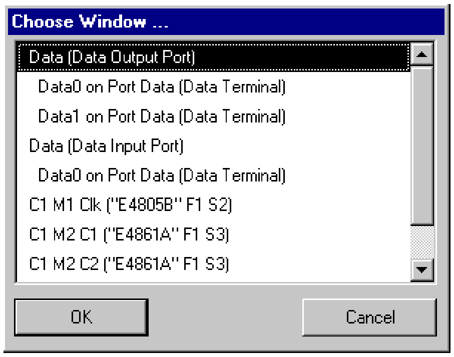
It should be obvious that the parameters of a connected channel are identical with the parameters of the terminal connected to that channel.

Alternatively, you can also choose *Parameter Editor* from the *Go* menu.

In this case, you get a list of all the items that have been configured so far and can have parameters. Such items are:

- the clock module
- data and pulse ports
- connected terminals (= connected channels)
- unconnected instrument channels

Select an item from the list to open the Parameter Editor with the respective view.



Choose Window ...

Data (Data Output Port)
 Data0 on Port Data (Data Terminal)
 Data1 on Port Data (Data Terminal)
Data (Data Input Port)
 Data0 on Port Data (Data Terminal)
C1 M1 Clk ("E4805B" F1 S2)
C1 M2 C1 ("E4861A" F1 S3)
C1 M2 C2 ("E4861A" F1 S3)

Figure 120 **Parameter Editor Selection Window**

The display of the Parameter Editor depends on the type of port or channel that has been selected. A selection list and arrow buttons in the upper right-hand corner of the editor window enable you to switch to the other items.

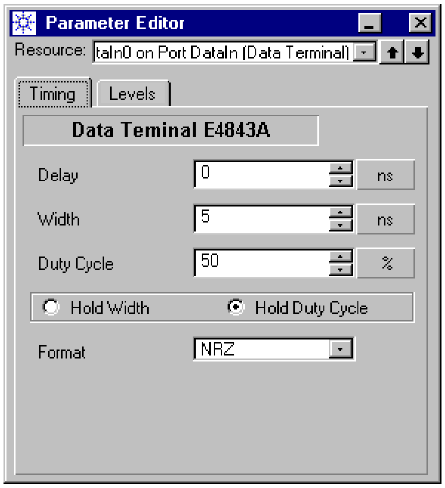
NOTE Port parameters refer to all channels connected to a port. Individual channel parameters override port parameters.

How to Set Up a DUT Input Port or Generator Channel

To set the parameters for a DUT port or terminal that receives generated signals or a channel that generates signals:

♦ Open the Parameter Editor for the port or channel (see *"How to Start the Parameter Editor for Ports/Channels" on page 224*).

Depending on the module or frontend type, the Parameter Editor window contains two or more pages. It starts with the *Timing* page.



Resource: taIn0 on Port DataIn (Data Terminal)

Data Teminal E4843A

Delay 0 ns

Width 5 ns

Duty Cycle 50 %

○ Hold Width ● Hold Duty Cycle

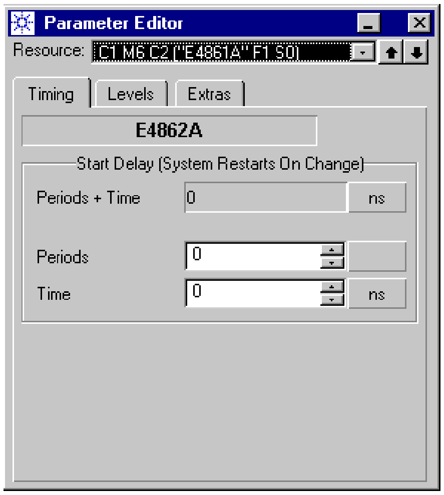
Format NRZ

Figure 121 Timing Parameters for an E4843A Data Generator

NOTE The window always displays the properties of the frontend channel(s). This implies, that the settings for a connected channel also affect the connected DUT terminal, and vice versa.

Note also that you have only limited access to the parameters of frontends which are connected to an E4868A MUX module (see *"Additional Characteristics of ParBERT 43G Systems" on page 116*).

Depending on the type of the frontend, the editor can contain different parameters than are shown in the figure above. For example, only the start delay can be set on the *Timing* page of an E4862A generator.



Resource: C1 M6 C2 ("E4861A" F1 S0)

Periods + Time 0 ns

Periods 0

Time 0 ns

Figure 122 Timing Parameters for an E4862A Data Generator

How to Set Generator Timing Parameters

To set the timing parameters for a generator channel:

1 Check the units.

The Parameter Editor displays default units and has default vernier steps. Both can be adjusted according to your needs (see *"How to Change Units and/or Vernier Steps" on page 148*).

2 Set the timing options:

– *Start Delay*:

The start delay is relative to the system clock start. You can specify a fixed time and/or a fraction of the system period. A negative delay can be set, if the system clock has been delayed (see *"How to Set the Clock Frequency" on page 177*).

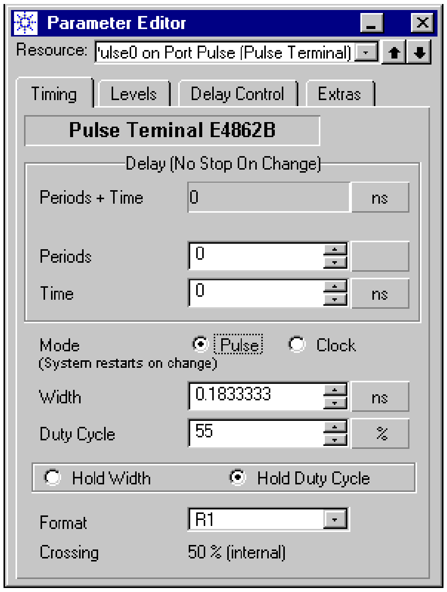
The *Periods + Time* field displays the actual delay.

– *Mode* (only with frontends of type E4862B, unconnected or connected to a pulse port):

All generator frontends that are connected to a pulse port, generate a clock signal with the frequency set on the Frequency page of the clock module. For the E4862B frontends, you can choose between *Pulse Mode* and *Clock Mode*.

If this parameter is changed while a test is running, the test is aborted and automatically restarted.

Pulse Mode is used for generating a clock signal with normal stability. A start delay can be specified, the data format, and also width or duty cycle.



Resource: 'ulse0 on Port Pulse (Pulse Terminal)

Pulse Teminal E4862B

Delay (No Stop On Change)

Periods + Time: 0 ns

Periods: 0

Time: 0 ns

Mode (System restarts on change): ● Pulse ○ Clock

Width: 0.1833333 ns

Duty Cycle: 55 %

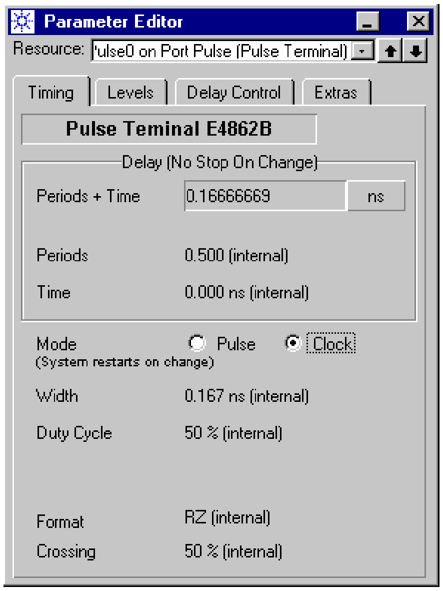
○ Hold Width ● Hold Duty Cycle

Format: R1

Crossing: 50 % (internal)

Figure 123 E4862B Data Generator Connected to Pulse Terminal in Pulse Mode

Clock Mode can be used for generating a very stable clock signal. In *Clock Mode*, all the additional timing parameters are locked, as illustrated below:



Resource: 'ulse0 on Port Pulse (Pulse Terminal)

Pulse Teminal E4862B

Delay (No Stop On Change)

Periods + Time: 0.16666669 ns

Periods: 0.500 (internal)

Time: 0.000 ns (internal)

Mode (System restarts on change): ○ Pulse ● Clock

Width: 0.167 ns (internal)

Duty Cycle: 50 % (internal)

Format: RZ (internal)

Crossing: 50 % (internal)

Figure 124 E4862B Data Generator Connected to Pulse Terminal in Clock Mode

In this mode, the frontend generates a precision pulse with 50 % duty cycle. The start delay (*Periods* + *Time*) that is displayed for your information, is calculated from the internal period delay, the delay of the system clock (if applicable), and the delays of zero adjustment and cable delay compensation.

– *Width* or *Duty Cycle*:

These parameters are mutually dependent. If you have typed a number into one of the two fields, terminate your input with the Return or Enter key. This updates the other field (*Width* or *Duty Cycle*).

– *Hold Width* or *Hold Duty Cycle*:

For the case the clock frequency is changed, you can set one of the two values to be fixed.

– *Rise/Fall Time* (with E4838A frontends only):

In the Parameter Editor, the rise/fall time can be set in the range of 0.5 to 4.5 ns.

The rise and fall times are coupled. The resulting signal shapes are illustrated below:

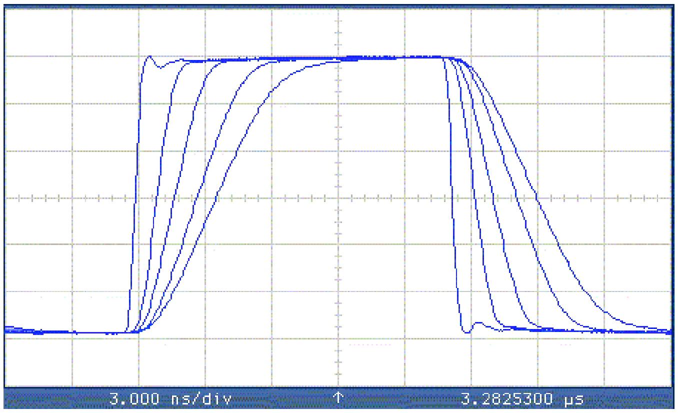


Figure 125 **Variable Slopes Setting**

– *Format*:

Choose from the list.

The default signal format for a data port and the channels connected to a data port is NRZ. The default format for a pulse port or an unconnected generator frontend is RZ.

In RZ format, every logical 1 generates a pulse, as illustrated in the figure below.

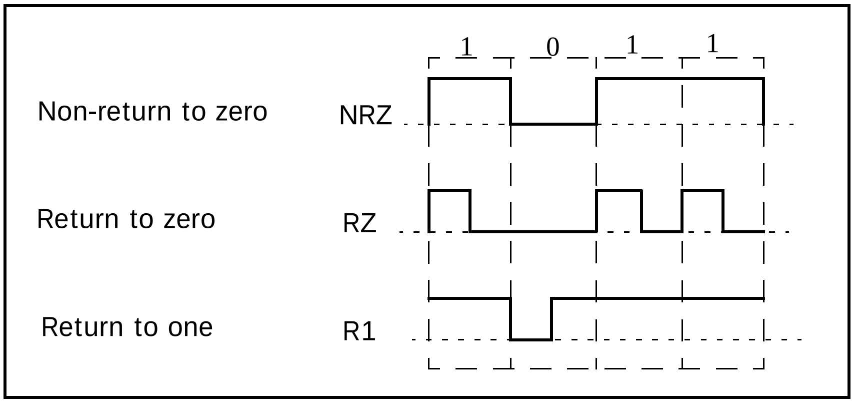
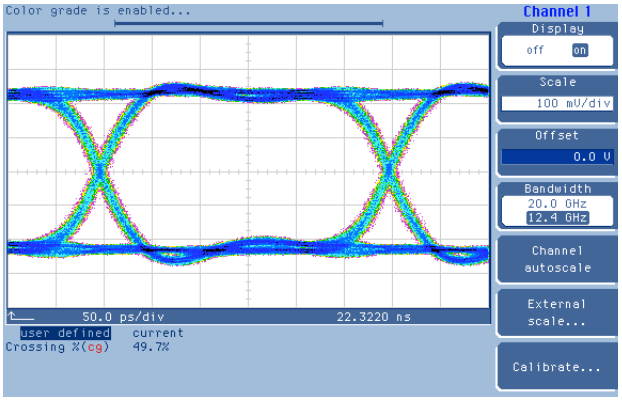


Figure 126 Signal Formats

- *Crossing* (only with E4862B frontends connected to data port terminals):

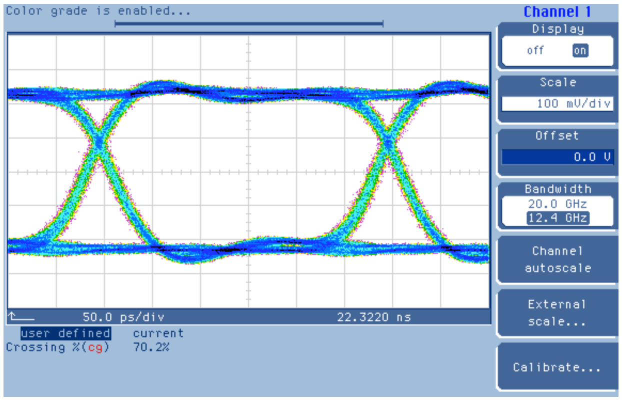
  If you have chosen the NRZ data format, you can shift the “crossing point” from 50 % of the signal amplitude down to 30 % or up to 70 %. This allows you to generate signals with non-symmetrical eye openings.

The following figures show two examples at 3.3 GHz.



Scale 100 mV/div
Offset 0.0 V
Bandwidth 20.0 GHz 12.4 GHz
50.0 ps/div 22.3220 ns
user defined current
Crossing %(cg) 49.7%

Figure 127 Crossing set to 50 %



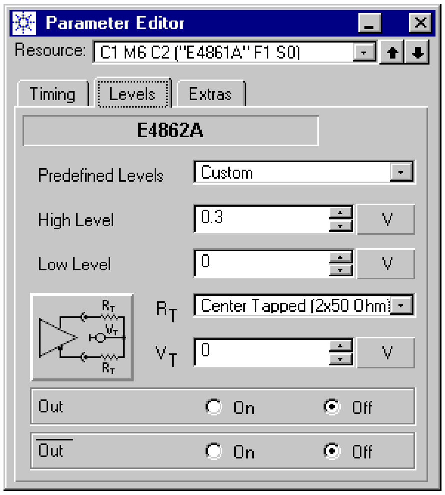
Scale 100 mV/div
Offset 0.0 V
Bandwidth 20.0 GHz 12.4 GHz
50.0 ps/div 22.3220 ns
user defined current
Crossing %(cg) 70.2%

Figure 128 Crossing set to 70 %

In RZ or R1 mode, or if the generator is connected to a pulse port, the crossing point is always at 50 %.

How to Set Electrical Generator Levels and Termination

1 In the Parameter Editor window, click the *Levels* tab.



Parameter Editor

Resource: C1 M6 C2 ("E4861A" F1 S0)

Timing | Levels | Extras

E4862A

Predefined Levels: Custom

High Level: 0.3 V

Low Level: 0 V

R_T: Center Tapped (2x50 Ohm)

V_T: 0 V

Out ○ On ● Off

$\overline{\text{Out}}$ ○ On ● Off

Figure 129 Level and Termination Parameters for a Connection With a Data Generator

2 Check the units.

The Parameter Editor displays default units and has default vernier steps.

Both can be adjusted according to your needs (see *"How to Change Units and/or Vernier Steps" on page 148*).

3 Set the signal high and low levels.

You can set your own custom levels or choose one of the *Predefined Levels*.

The predefined levels for a generator depend on the frontend.

They are listed in the following table:

Table 17 Predefined Generator Signal Levels

Name	High Level	Low Level	Termination Voltage	Termination Impedance	E4838A	E4843A	E4862A E4864A	E4862B E4810A	E4866A
TTL (into open)	2.5 V	0.0 V	0.0 V	Open		x			
TTL (into 50 Ω to GND)	2.5 V	0.0 V	0.0 V	2 × 50 Ω	x	x	x		
CMOS 5V (into open)	5.0 V	0.0 V	0.0 V	Open		x			
CMOS 3.3V (into open)	3.3 V	0.0 V	0.0 V	Open		x			
ECL (into 50 Ω to -2V)	-0.9 V	-1.7 V	-2.0 V	2 × 50 Ω	x	x	x	x	x
ECL (into 50 Ω to GND)	-0.9 V	-1.7 V	0.0 V	2 × 50 Ω	x	x	x	x	x
PECL (into 50 Ω to 3V)	4.1 V	3.3 V	3.0 V	2 × 50 Ω	x	x	x		
PECL (into 50 Ω to 1.3V)	2.4 V	1.6 V	1.3 V	2 × 50 Ω	x	x	x	x	x
CML (–0.4 to GND)[a]	0 V	–0.4 V	0.0 V	2 × 50 Ω			x	x	

[a] The CML level is used when the frontend is connected to an E4868A/B MUX module.

4 Set the expected signal termination.

Termination impedance is usually the input impedance of the DUT.

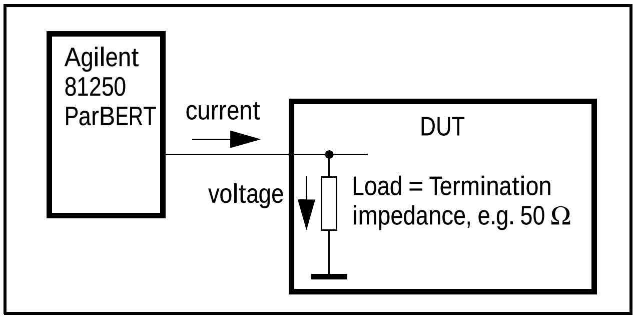


Figure 130 Termination Impedance (single ended)

The *termination voltage* for a terminal that is not connected to ground must not remain below –2 V.

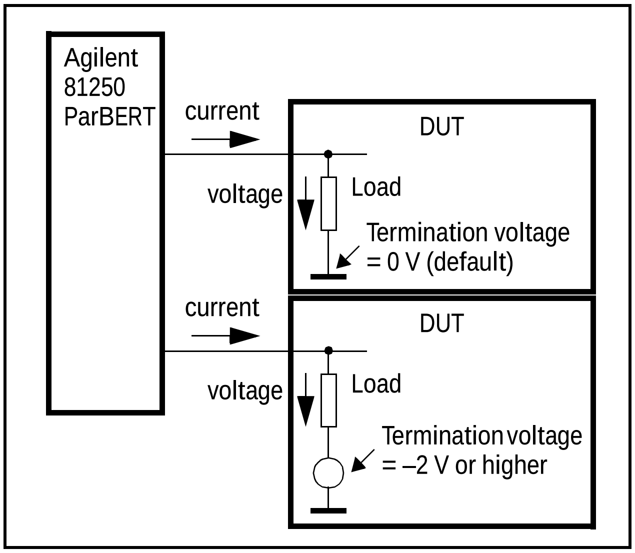


Figure 131 Termination Voltage (single ended)

Special load options are supported by the differential generator frontends E4838A, E4843A, E4862A/B, and E4864A/B. For these frontends, you can choose between center tapped and differential termination by clicking the graphical button.

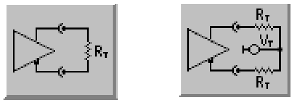


Figure 132 Termination Options for Differential Generators

Center tapped termination uses two 50 Ω resistors. The termination voltage range is –2 to +3 V.

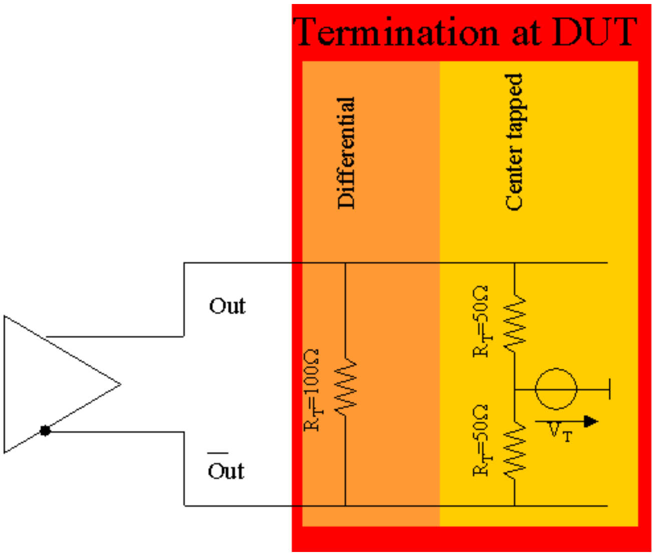


Figure 133 Supported Signal Termination Options at the DUT

5 Choose the polarity (E4838A frontend only): This allows to invert the polarity of the output signal.

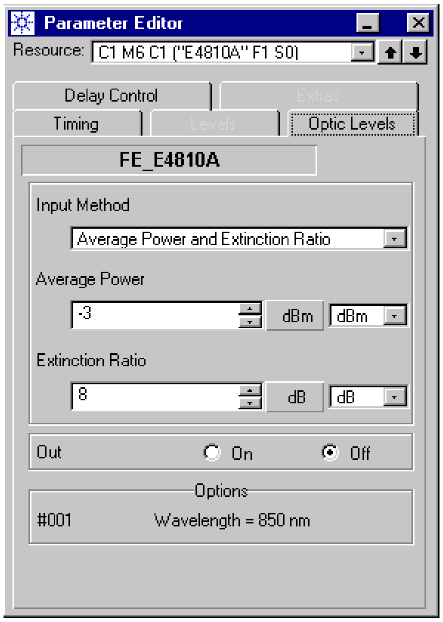
6 Activate the connection by switching the output radio button to *On*. This is probably the most important step of all!

TIP When running a test, check the green LEDs on the frontends. They indicate whether the channel is enabled or disabled. If a channel is disconnected due to hardware constraints, correct its physical termination and then use the Connectors On/Off button to re-establish the connection.

How to Set Optical Generator Levels

Optical generator levels can be set for optical DUT input ports or terminals and generators in optical mode.

1 In the Parameter Editor window, click the *Optic Levels* tab.



Resource: C1 M6 C1 ("E4810A" F1 S0)

Input Method: Average Power and Extinction Ratio

Average Power: -3 dBm dBm

Extinction Ratio: 8 dB dB

Out ○ On ● Off

Options

#001 Wavelength = 850 nm

Figure 134 Optical Level Parameters

2 Choose the *Input Method*. Your choice has an impact on the parameters that can be set.

Choices are:

- *Average Power and Extinction Ratio*
- *Average Power and Modulation Amplitude*
- *High Power and Low Power*

The parameters are defined as shown in the following figure:

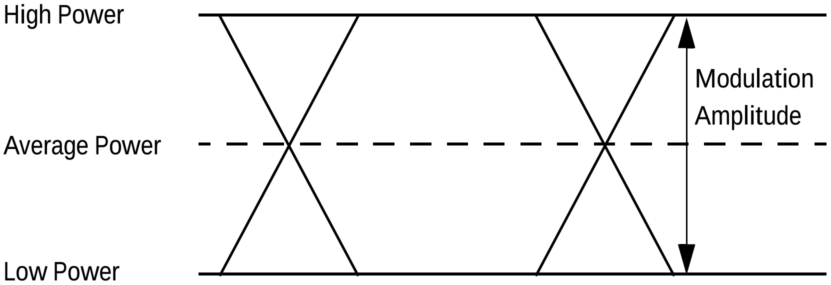


Figure 135 **Definition of Optical Level Parameters**

Average Power:

```
AP[W] = (HiP[W] + LoP[W]) / 2
```

Modulation Amplitude:

```
MA[W] = HiP[W] – LoP[W]
```

Extinction Ratio:

```
ER[-] = HiP[W] / LoP[W]
ER[dB] = 10 log(HiP[W] / LoP[W])
```

3 Check and if necessary change the units.

For power values, you can switch between dBm and Watt (mW).

$P[dBm] = 10\ \log(P[W]/1mW)$

$P[W] = 1mW \times 10^{(P[dBm]/10)}$

For the extinction ratio, you can switch between dB and the unitless fraction.

4 Set the signal levels.

5 Activate the laser by switching the output radio button to *On*.

TIP Check the green LEDs on the frontends. They indicate whether a channel is enabled or disabled.

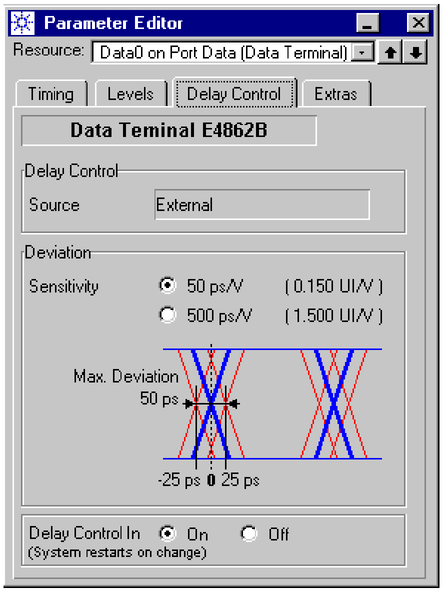
NOTE The optical levels you can set with the Parameter Editor refer to the output of the built-in laser source. Please refer to the technical specifications for calibration conditions and data.

The optical levels seen by the DUT depend much on additional factors like bit rate, pattern, connector uncertainty, and fiber characteristics.

It is therefore recommended to add an optical oscilloscope to the test setup. This allows you to observe and measure the eye opening at the DUT under test conditions. You can then fine-tune the level parameters with the Parameter Editor.

How to Use the Variable Delay (FEs of Type E4862B Only)

The generator frontends of type E4862B have an input connector that allows to vary the signal delay by applying a voltage. This input is controlled from the following page:



Resource: Data0 on Port Data (Data Terminal)

Data Terminal E4862B

Delay Control

Source External

Deviation

Sensitivity ● 50 ps/V (0.150 UI/V) ○ 500 ps/V (1.500 UI/V)

Delay Control In ● On ○ Off
(System restarts on change)

Figure 136 **Delay Control Parameters for E4862B Data Generators**

Acceptable input voltages are between –0.5 V and +0.5 V. Sweep frequencies from DC to 200 MHz may be applied.

To use the delay input:

1 Set the sensitivity. Choices are 50 ps/V or 500 ps/V.

   For example, 50 ps/V gives you a sweep range from –25 ps to +25 ps.

   For your information, the sensitivity is also displayed in UI/V. One UI (Unit Interval) corresponds to the current system clock period. In the figure above, the system clock period was set to 333.333 ps (3.0 GHz). Hence, a sensitivity of 50 ps/V yields a total sweep range of 0.15 UI/V.

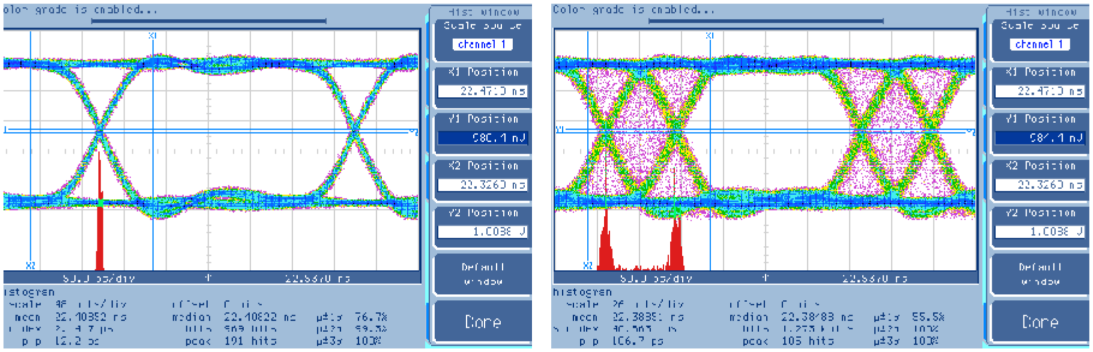
   This is acceptable—the full voltage swing of ±0.5 V can be used.

   Care should be taken if the chosen sensitivity exceeds 0.4 UI/V. In this case, the full voltage swing of ±0.5 V may have the effect that the signal cannot be captured and analyzed.

2 Switch the delay control input *On*. A green LED on the frontend indicates the status.

If this is done while a test is running, the test is aborted and automatically restarted.

The following figure shows an example measured at 3.3 GHz:



"Clean Eye" @ 3.3Gbit/s

Jitter modulated with Rectangle-Wave

Figure 137 **Delay Modulation (Voltage-Controlled Jitter)**

Below the eye opening, you can see the jitter distribution. A rectangle control signal generates two peaks. Using triangle waveforms, sine waves, or arbitrary control signals, you can simulate all kinds of jitter distribution.

How to Set the Global Disconnect Mode

If you have selected a port or channel that supports data rates above 675 Mbit/s, then the Parameter Editor shows an additional tab: *Extras*.

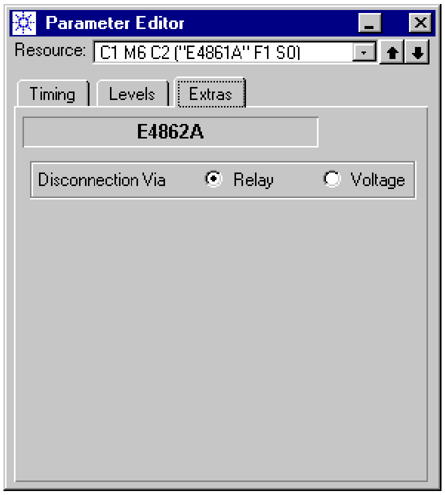
This page allows you to change the default behavior of the global Connectors Off/On button of the tool bar.

This button is used for disconnecting all frontends from the DUT. It is generally clicked before changing the DUT. Once the new DUT is mounted, it is clicked once more to re-establish all the previous connections.

The same functions are also provided by the *Control* menu.

By default, *Connectors Off* opens all input and output relays. It is now possible to specify whether the relays shall be switched or whether the frontends shall be disconnected by grounding.

Especially in a production environment, grounding is a way to increase the lifetime of the ParBERT relays.



Resource: C1 M6 C2 ("E4861A" F1 S0)

Disconnection Via ● Relay ○ Voltage

Figure 138 Extras Page for a Data Generator

The default is *Disconnection via Relay*.

Disconnection via Voltage means:

- For generators: The output voltage is set to 0 V.
- For analyzers: The comparator and termination voltages are set to 0 V.

You have to check whether this is in harmony with your DUT. The ParBERT generators and analyzers are connected via 50 Ω resistors. Currents will flow, if the DUT pins have not 0 V. You should also consider the possibility of static discharge.

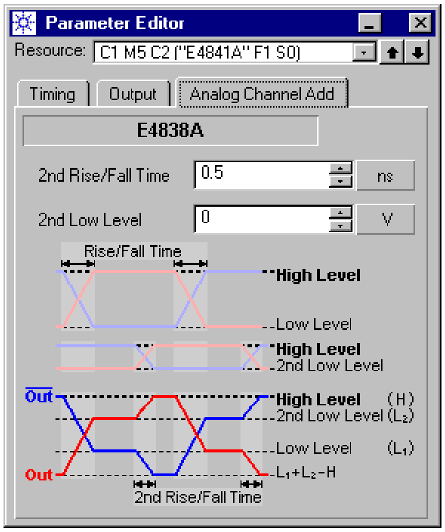
NOTE This setting impacts only the global Connectors Off/On function. The relays are always switched, if you enable or disable connectors with the Parameter Editor.

How to Add Channels in Analog Mode

If you have selected an E4838A frontend, the Parameter Editor shows an additional tab: *Analog Channel Add*. This page provides additional parameters for combining two output channels.

NOTE These parameters are just set with the Parameter Editor. To activate them, you need to combine the channel with the channel above in the Channel Configuration Editor (see *“How to Combine Generator Channels” on page 251*).

An output channel of an E4838A frontend can only be combined with the channel above if this other channel is also an E4838A or an E4843A frontend.



Parameter Editor

Resource: C1 M5 C2 ("E4841A" F1 S0)

Timing | Output | Analog Channel Add

E4838A

2nd Rise/Fall Time: 0.5 ns

2nd Low Level: 0 V

Figure 139 **Analog Channel Add Parameters**

The window illustrates how the analog channel addition works.

You are adding two differential signals. A differential signal consists of an OUT and a complementary \OUT\ voltage. Both signals can have different slew rates (rise/fall times) and amplitudes (the voltage difference between high and low level).

But the output voltage of the E4838A frontend at 50 Ω load is limited to –2.2 V to +4.4 V, and the output voltage swing must be in the range of 0.05 V to 3.5 V (for details refer to the *Agilent 81250 Technical Specifications*).

So we keep the high level voltage of the frontend that holds the connector. That means, the connected channel's high level voltage is only reached after adding the two OUT channels. The amplitudes Amp1 and Amp2 remain unchanged, and the resulting voltage swing of the signal is the sum of both amplitudes.

A similar scheme applies to the complementary \OUT\ connector.

To change the characteristics of the second channel:

1 Adjust the rise/fall time of the second channel (only E4838A or E4843A frontends, see also *"How to Set Generator Timing Parameters" on page 227*).

2 Set the low level of the second channel. This specifies the amplitude Amp2.

Error messages will be displayed if you try to override the physical limits of the E4838A frontend. If that occurs, you need to correct the amplitudes.

Example

Assume the following setting:

High level = 1.5 V

First low level = 0 V

Second low level = 0 V

This results in the following OUT signal:

Table 18 **Output Voltage for Analog Channel Addition**

Logic at Channel 1	Logic at Channel 2	Output Voltage
1	1	1.5 V
1	0	0 V
0	1	0 V
0	0	–1.5 V

The method is once more illustrated below:

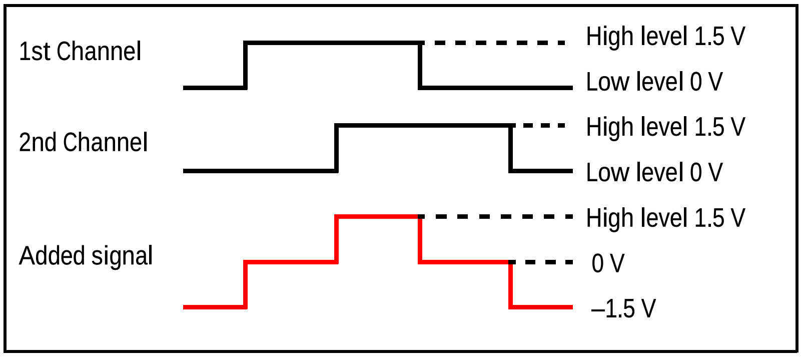


Figure 140 **Analog Channel Addition**

The above setting is accepted because the resulting amplitude and voltages can be generated by the E4838A frontend.

If you have chosen one of the predefined levels for the connected generator channel, you may need to change the second low level which is set to 0 V by default.

TTL into 50 Ω, for example, would yield a voltage swing of 5 V, which is unacceptable. But if you change the second low level to for instance 2 V (for generating spikes of 0.5 V), this is well in the range of the E4838A frontend.

A voltage overshot could hence be generated as illustrated below.

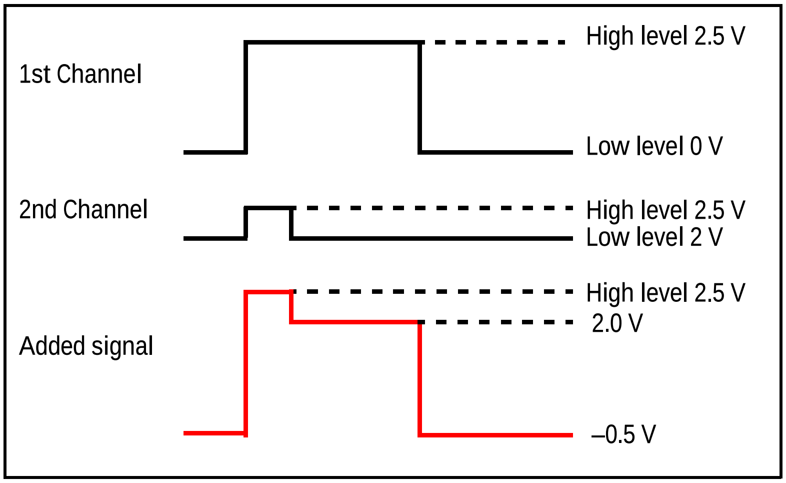


Figure 141 **Overshot Generation**

To generate real-world signals, you will also have to take the rise/fall times into account.

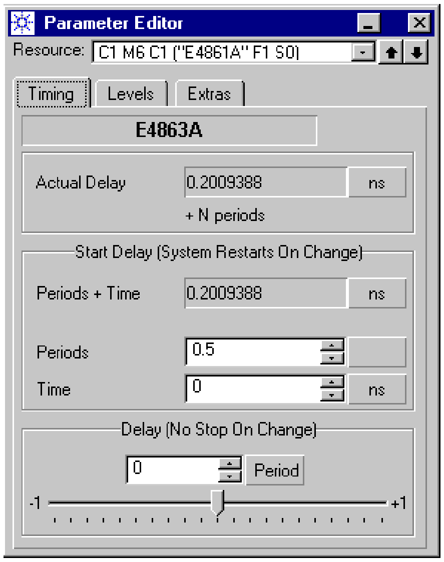
NOTE The Analog Channel Add setting only takes effect after the two channels have been added by means of the Channel Configuration Editor (see *“How to Combine Generator Channels” on page 251*).

Both channels can have different timing parameters, such as frequency, pulse width and delay. High level and expected load are determined by the channel that holds the connector.

How to Set Up a DUT Output Port or Analyzer Channel

To set the parameters for a port or terminal that returns signals or for a channel that receives signals:

1 Open the Parameter Editor for the port or channel (see *"How to Start the Parameter Editor for Ports/Channels" on page 224*).



Resource: C1 M6 C1 ("E4861A" F1 S0)

E4863A

Actual Delay 0.2009388 ns
+ N periods

Start Delay (System Restarts On Change)

Periods + Time 0.2009388 ns

Periods 0.5

Time 0 ns

Delay (No Stop On Change)

0 Period

-1 +1

Figure 142 Timing Parameters for a Data Analyzer

NOTE The Parameter Editor always displays the identification of the connected frontend. Thus, changes done in the channel parameters will also appear in the settings of the connected port terminal and vice versa.

Note also that you have only limited access to the parameters of frontends which are connected to an E4869A DEMUX module (see *"Additional Characteristics of ParBERT 43G Systems" on page 116*).

How to Set Analyzer Timing Parameters

1 Check the units.

The Parameter Editor displays default units and has default vernier steps. Both can be adjusted according to your needs (see *“How to Change Units and/or Vernier Steps” on page 148*).

2 Set the timing options:

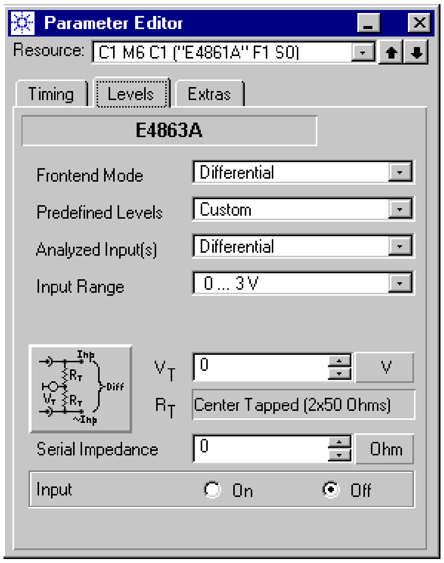
– *Periods and Time*: The total delay for capturing received data is composed of a relative delay (in fractions of system clock cycles) and an absolute delay (independent of the system clock).

If one of these parameters is changed while a test is running, the test is aborted and restarted.

– *Delay (No Stop On Change)*: The sampling delay of analyzer frontends can be fine-tuned within up to ±1 clock periods. The phase can be adjusted with the vernier without interrupting a running test. See also *“Manual Analyzer Sampling Delay Adjustment” on page 88*.

How to Set Analyzer Levels and Termination

1 Click the second tab of the analyzer parameter window. It is labeled *Levels*



Parameter Editor

Resource: C1 M6 C1 ("E4861A" F1 S0)

E4863A

Frontend Mode: Differential

Predefined Levels: Custom

Analyzed Input(s): Differential

Input Range: 0 ... 3 V

V_T: 0 V

R_T: Center Tapped (2x50 Ohms)

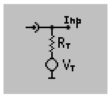
Serial Impedance: 0 Ohm

Input ○ On ● Off

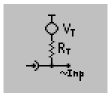
Figure 143 **Additional Parameters for Data Analyzers**

2 Select the frontend mode. Choices are:

- *Differential*
- *Single-ended Normal*



- *Single-ended Complement*



3 Decide on the additional options. The additional options are:

- A set of *Predefined Levels*: Choose one of the predefined levels, use the defaults values or specify your own settings using *Custom* in this field.

  The predefined levels for analyzers are listed in the table below:

Table 19 Predefined Analyzer Signal Levels

Name	Threshold	Termination Voltage	Internal Impedance	E4835A	E4863A E4865A E4863B E4811A	E4867A
ECL (to –2V)	-1.3 V	-2.0 V	50 Ω	x	x	x
ECL (to GND)	-1.3 V	0.0 V	50 Ω	x	x	x
PECL (to +3V)	3.7 V	3.0 V	50 Ω	x		
TTL (to GND)	1.5 V	0.0 V	50 Ω	x	x	

- If you are using your own custom setting:

  The minimum *Termination Voltage* for a terminal that is not connected to ground is –2 V.

  The standard input *Impedance* of an analyzer frontend is 50 Ω. See the specifications for details.

4 The *Serial impedance* is usually the output impedance of the DUT.

5 Do not forget to activate the connection by switching the *Input* button to *On*.

NOTE When running a test, check the green LEDs on the frontends. They indicate whether the channel is enabled or disabled.

Differential Analyzer Frontends

Special options are available for the differential analyzer frontends E4835A, E4837A, E4863A/B, and E4865A.

For these frontends you can:

- Select the *Analyzed Input(s)*. Choices are: Normal input IN, complementary input \IN\, differential input.
- If the **frontend mode** is set to *Differential* (not the input type!), you can choose between center tapped or differential input signal termination by clicking the graphical button.

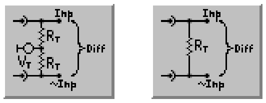


Figure 144 Termination Options for Differential Analyzers

Center tapped termination uses two 50 Ω resistors. The termination voltage range is –2 V to +3 V. Differential termination inserts a 100 Ω resistor.

Hence, these frontends provide the following input and measurement options:

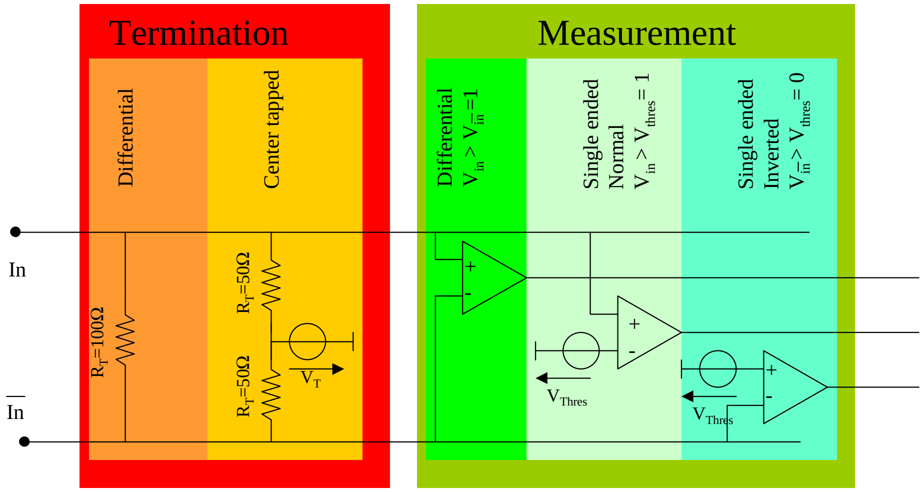

Termination
Differential
Center tapped
Measurement
Differential $V_{in} > V_{\overline{in}} = 1$
Single ended Normal $V_{in} > V_{thres} = 1$
Single ended Inverted $V_{in} > V_{thres} = 0$
In
$\overline{In}$
R_T=100Ω
R_T=50Ω
R_T=50Ω
V_T
V_{Thres}
V_{Thres}
+
-
+
-
+
-

Figure 145 Termination and Measurement Options for Differential Analyzers

- For the analyzer frontends E4835A and E4837A, you can invert the signal polarity with the *Polarity* radio button.

  The settings of *Input Type* and *Polarity* have the following effects on data recognition:

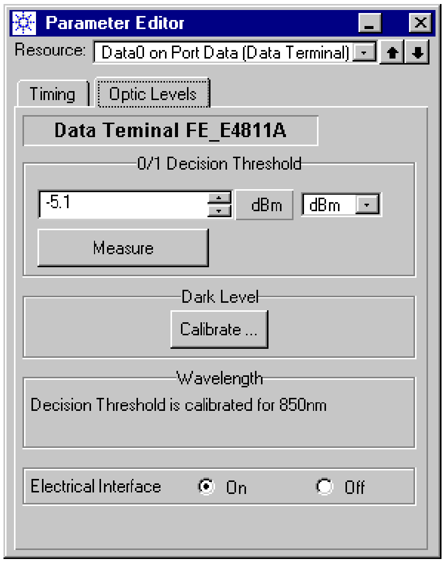
Table 20 Data Acquisition with a Differential Analyzer

Input Type	Input Signal Condition	Acquired Data	
		Normal Polarity	Inverted Polarity
Differential	Signal at IN > \IN\	1	0
	Signal at IN < \IN\	0	1
Normal Input	Signal at IN > Threshold	1	0
	Signal at IN < Threshold	0	1
Complementary Input	Signal at \IN\ > Threshold	0	1
	Signal at \IN\ < Threshold	1	0

How to Set Optical Analyzer Levels

Optical analyzer levels can be set for optical DUT output ports or terminals and analyzers in optical mode.

1 In the Parameter Editor window, click the *Optic Levels* tab.


Parameter Editor

Resource: Data0 on Port Data (Data Terminal)

Data Terminal FE_E4811A

0/1 Decision Threshold: -5.1 dBm

Wavelength: Decision Threshold is calibrated for 850nm

Electrical Interface ● On ○ Off

Figure 146 Optical Level Parameters

2 Set the *0/1 Decision Threshold*.

This setting refers to the comparator built into the O/E converter (see also *"E4810A/E4811A Modules" on page 46*). For optical signals, the threshold for the decision between zero and one is defined in power units.

You can set the threshold in dBm or mW.

$$P[dBm] = 10 \log(P[W]/1mW)$$

$$P[W] = 1mW \times 10^{(P[dBm]/10)}$$

If an input signal is present, you can also click the Measure button. When this is done, the average signal power is measured and used as threshold.

NOTE The power meter consists of a low pass filter and an ADC.

If the incoming signal has an equal distribution of bright and dark levels (like, for example, a sequence of 0, 1, 0, 1, 0, 1 ...), the threshold will be set to:

$$AP[W] = (HiP[W] + LoP[W]) / 2$$

where

AP = Average power

HiP = High power

LoP = Low power

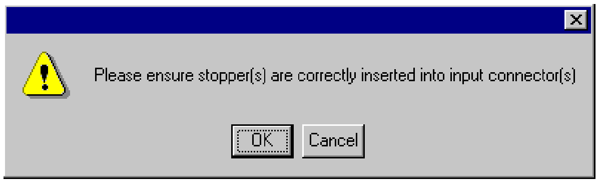
If the incoming signal has long runs of bright or dark levels, the measured power level may be inadequate and the threshold has to be set manually.

3 Calibrate the *Dark Level*.

Optical sensors exhibit a drift with temperature. Dark level calibration compensates for that drift (for 850 nm optical wavelength).

Dark level calibration should be performed after warming up the system (not before half an hour after power on). It should also be performed if the room temperature has changed drastically (by more than 10 degrees).

For dark level calibration, the fiber must be disconnected and the optical input covered. The plastic cap delivered with the module should be used. This is meant by the following message:



Please ensure stopper(s) are correctly inserted into input connector(s)

Figure 147 Dark Level Calibration Reminder

If you have chosen a terminal or analyzer channel, this message refers to that terminal or channel. If you have chosen a port, this message refers to all terminals of that port.

After the optical connector has been covered, click *OK*.

NOTE You can also choose *Dark Level Calibration* from the *System* menu. With that function, you can can either calibrate all installed optical analyzers in parallel or move from one optical analyzer to the next.

The parallel procedure requires that all analyzer connectors are covered.

The sequential procedure stops before each measurement, so that you can attach the cap on the optical connector that is going to be calibrated.

4 Activate the O/E converter by switching its *Electrical Interface* to *On*.

TIP Check the green LEDs on the frontends. They indicate whether a channel is enabled or disabled.

How to Combine Generator Channels

Combining generator channels enables you to test devices with multiple edge timings or multi-level signals.

NOTE The channel add function enables you to combine two or four generator channels. The channels to be added must all reside in one module and must be contiguous. Adding channels reduces the number of active connectors.

Digital channel addition The digital channel addition is an XOR addition (exclusive OR or modulo 2 addition). For details see the figure below. The addition takes place before levels are applied to the signals. The final signal is routed to one output amplifier.

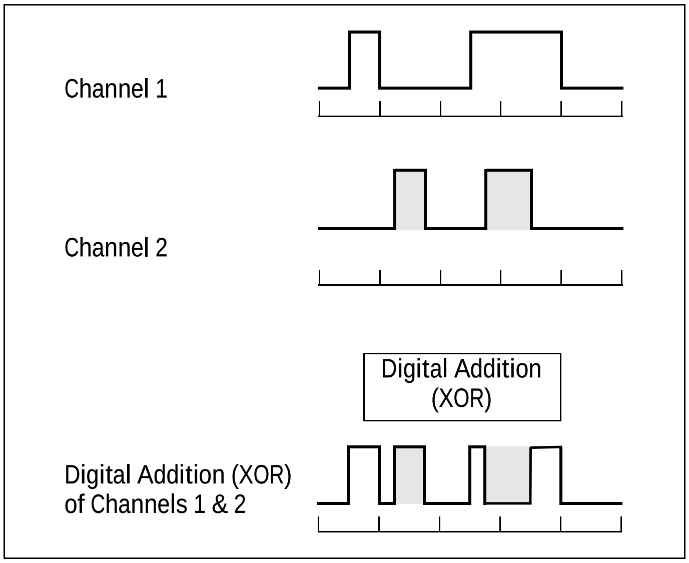


Figure 148 **Digital Channel Addition**

Analog channel addition The analog channel addition allows to add signal voltages. It is available if the module contains at least one E4838A generator module. You can thus produce signals or pulses with overshot, ringing, and so on.

Channels are added with the Channel Configuration Editor.

How to Start the Channel Configuration Editor

The Channel Configuration Editor is started from the Connection Editor.

To start the Channel Configuration Editor:

1 Double-click the configuration area of a generator frontend.

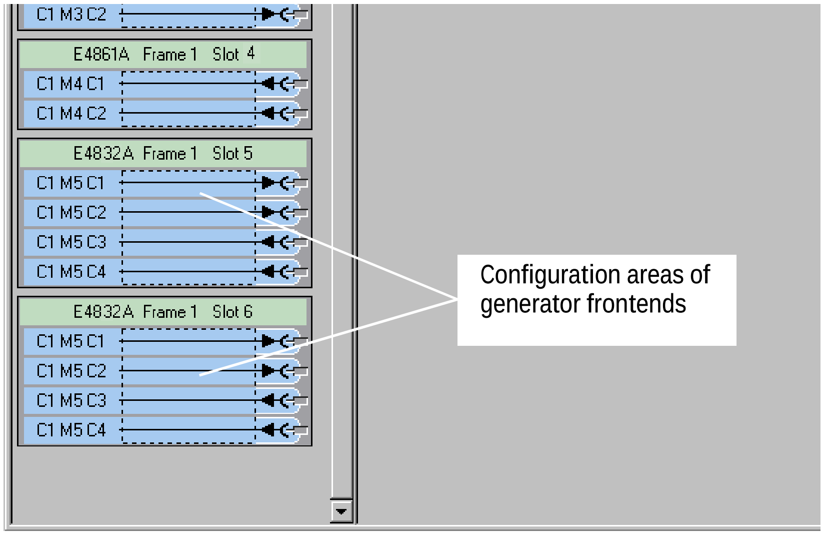
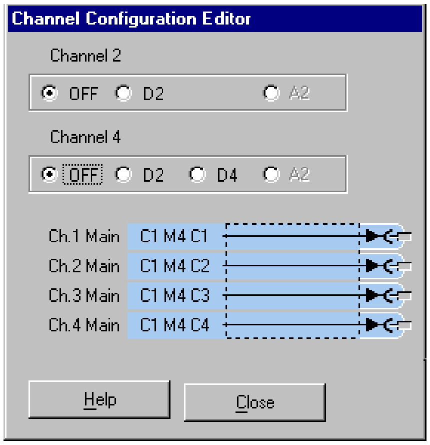


Figure 149 Module Configuration Area

Alternatively, you can also click on the configuration area with the right mouse button and select *Properties*.

How to Use the Channel Configuration Editor

The Channel Configuration Editor comes up with the following window:



Channel Configuration Editor

Channel 2

● OFF ○ D2 ○ A2

Channel 4

● OFF ○ D2 ○ D4 ○ A2

Ch.1 Main	C1 M4 C1
Ch.2 Main	C1 M4 C2
Ch.3 Main	C1 M4 C3
Ch.4 Main	C1 M4 C4

Figure 150 Channel Configuration Editor

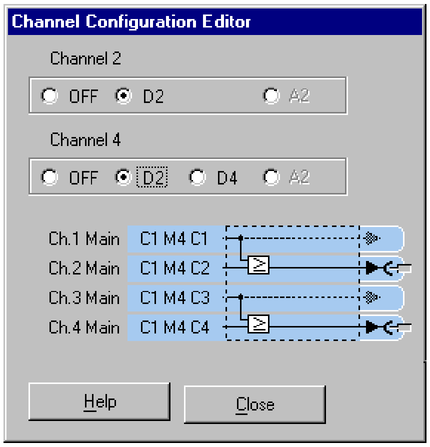
With the options of the Channel Configuration Editor, you can affect the properties of channel 2 and/or channel 4.

Combining Channels in Digital Mode

To add channels 1 and 2 and channels 3 and 4 in digital mode:

1 Activate D2 for channel 2.

2 Activate D2 for channel 4.



Channel Configuration Editor

Channel 2

○ OFF ● D2 ○ A2

Channel 4

○ OFF ● D2 ○ D4 ○ A2

Ch.1 Main	C1 M4 C1
Ch.2 Main	C1 M4 C2
Ch.3 Main	C1 M4 C3
Ch.4 Main	C1 M4 C4

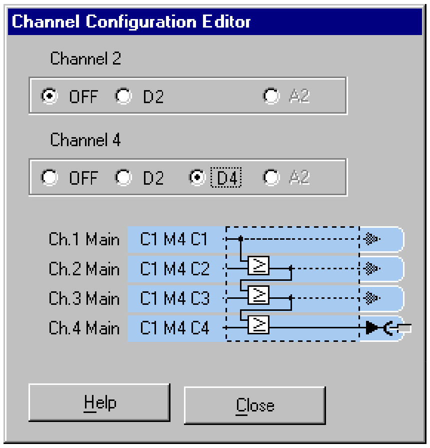
Figure 151 Channel Configuration Editor—Two Added Channels

Note that the connectors of channel 1 and channel 3 are no longer available for connections.

Combining Four Channels

To combine all four channels:

♦ Activate D4 for channel 4.



Channel Configuration Editor

Channel 2

● OFF ○ D2 ○ A2

Channel 4

○ OFF ○ D2 ● D4 ○ A2

Ch.1 Main C1 M4 C1

Ch.2 Main C1 M4 C2

Ch.3 Main C1 M4 C3

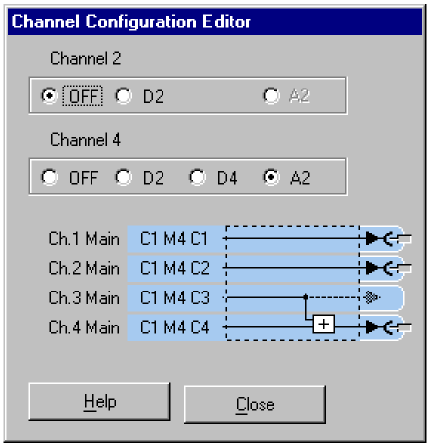
Ch.4 Main C1 M4 C4

Figure 152 Channel Configuration Editor—Four Added Channels

Note that the connectors of channels 1 to 3 are no longer available for connections.

Combining Channels in Analog Mode

If the module contains one or several E4838A generator frontends in even-numbered slots (channel 2 or 4) and identical or E4843A frontends in the slots above, you can also add two channels in analog mode by clicking the A2 radio button.



Channel Configuration Editor

Channel 2

● OFF ○ D2 ○ A2

Channel 4

○ OFF ○ D2 ○ D4 ● A2

Ch.1 Main C1 M4 C1

Ch.2 Main C1 M4 C2

Ch.3 Main C1 M4 C3

Ch.4 Main C1 M4 C4

Figure 153 Channel Configuration Editor—Analog Added Channels

If this option is selected, the parameter settings of the E4838A frontend for “Analog Channel Add” take effect (see *“How to Add Channels in Analog Mode” on page 240*).

How to Set Up N4868A Booster Channels

The N4868A 10.8 Gbit/s Booster Module is an add-on to the E4866A 10.8 Gbit/s data generator module.

It is used to reduce the transition times of the generated signals and can be connected between the E4866A data generator and the DUT.

For a hardware description, see *“N4868A Module” on page 51*.

The operating mode and parameters can be set in the Connection Editor. There, the module appears as shown below:

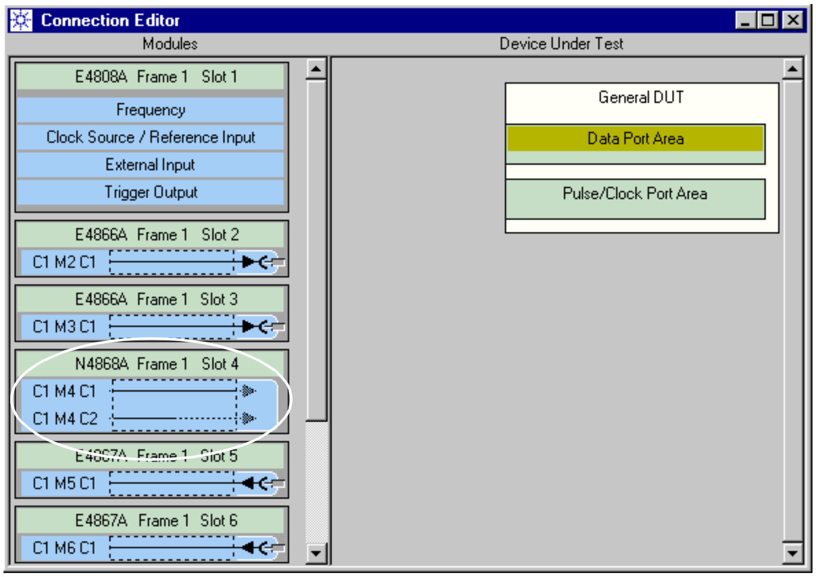


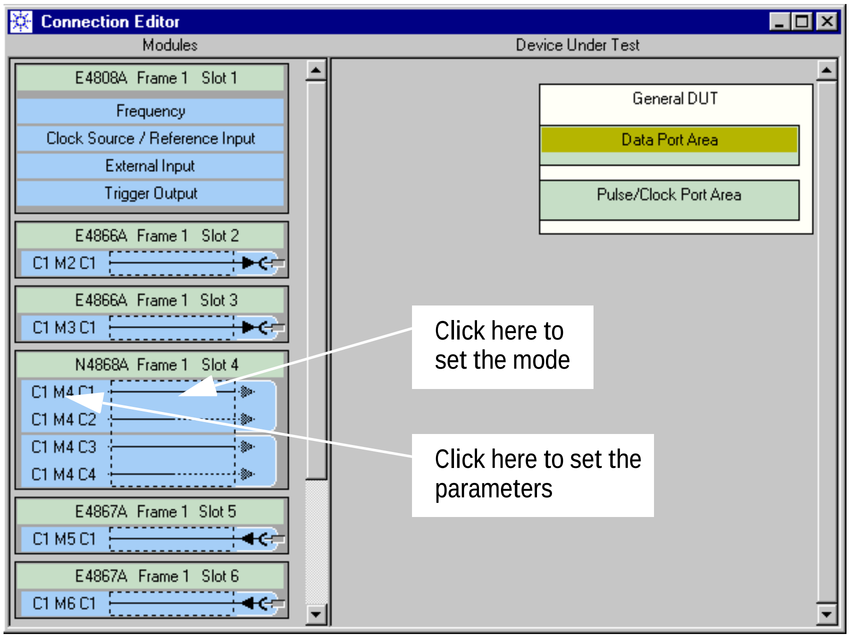
Figure 154 Representation of the Booster Module

NOTE Note that the symbol of the booster module has no “connectors” that could be dragged with the mouse.

The logical connections in the Connection Editor have to be made between the data generator modules and the terminals of the DUT input port.

Physical cable connections have to be made between the data generator module(s) and the booster module, and also between the booster module and the DUT.

The following figure shows a booster module with two frontends (four amplifiers):



Click here to set the mode

Click here to set the parameters

Figure 155 Booster Module with Opt. #001

The channel modes and parameters can be easily accessed from the Connection Editor (see the figure above).

How to Set the Operating Mode of the Booster Module

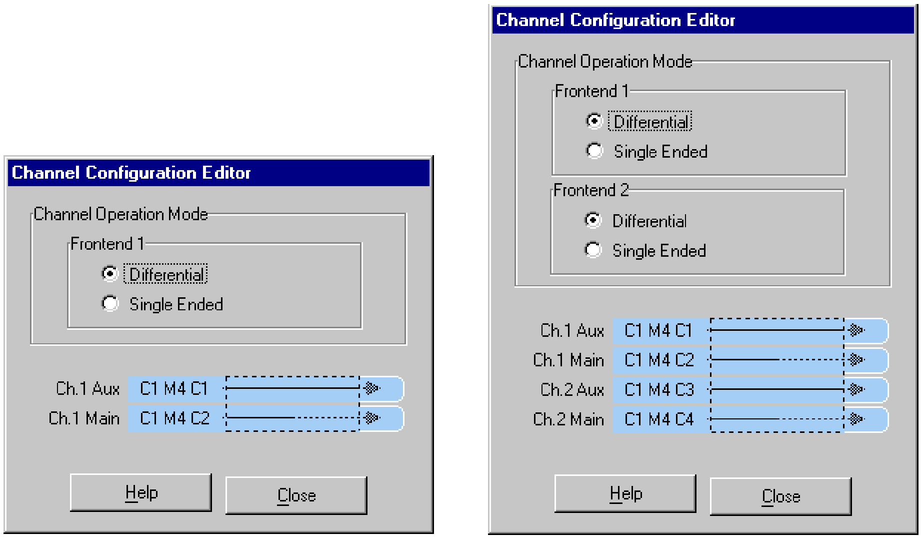
The operating mode of the booster module is set with the Channel Configuration Editor.

1 In the Connection Editor, double click the channel configuration area

or

right-click the channel configuration area and choose *Properties*.

This opens the Channel Configuration Editor for the module.



Channel Configuration Editor

Channel Operation Mode

Frontend 1

● Differential

○ Single Ended

Ch.1 Aux C1 M4 C1

Ch.1 Main C1 M4 C2

Channel Configuration Editor

Channel Operation Mode

Frontend 1

● Differential

○ Single Ended

Frontend 2

● Differential

○ Single Ended

Ch.1 Aux C1 M4 C1

Ch.1 Main C1 M4 C2

Ch.2 Aux C1 M4 C3

Ch.2 Main C1 M4 C4

Figure 156 Setting the Booster Module Mode

2 Enable *Differential* or *Single Ended* mode.

- *Differential* is the default mode. It requires a differential input signal. The output signal is also differential.
- In *Single Ended* mode, you can operate each amplifier separately. For example, you can connect two 10.8 Gbit/s generators—both set to single-ended operation—and amplify two separate signals.

Note that the symbol indicates the current mode.

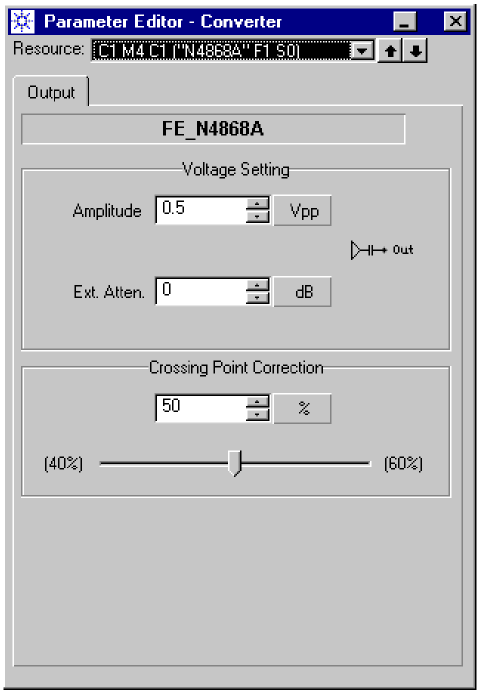
How to Set the Parameters of the Booster Module

The parameters of the booster channels are set with the Parameter Editor.

NOTE You can set individual parameters for every booster channel. They are all considered in single-ended mode. In differential mode, the parameters of every second booster channel (corresponding to amplifier B) are ignored.

1 In the Connection Editor, right-click the channel identifier (for example C1 M4 C1) and choose *Properties*.

This opens the Parameter Editor for this channel.



Parameter Editor - Converter

Resource: C1 M4 C1 ("N4868A" F1 S0)

Output

FE_N4868A

Voltage Setting

Amplitude 0.5 Vpp

Ext. Atten. 0 dB

Crossing Point Correction

50 %

(40%) (60%)

Figure 157 Setting the Booster Module Parameters

2 Set the output voltage amplitude.

This is the voltage swing to be applied to the DUT.

The output is AC coupled, as illustrated in the window. Therefore, only the amplitude (peak-to-peak voltage) can be specified.

The full voltage range without attenuator is 0.5 V to 2.5 V.

3 If you are using an external attenuator, enter its setting.

An external attenuator can be used to reduce the signal amplitude without increasing the signal-to-noise ratio.

The following figure illustrates its connection.

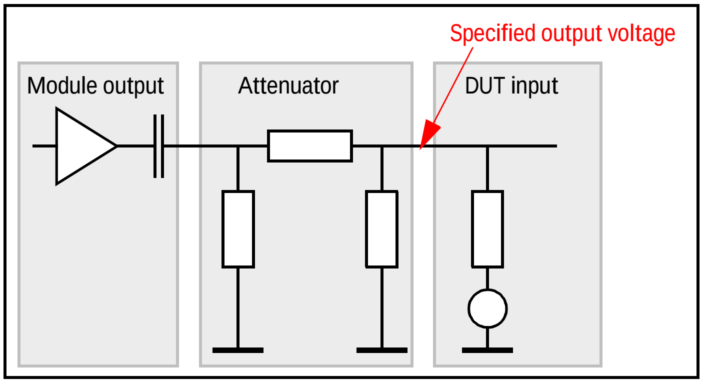


Figure 158 **Using an External Attenuator**

If you have specified an external attenuator, the range of the amplitude voltage is recalculated. The software considers the attenuator as a part of the channel which reduces the amplitude range.

For example, if you have set the attenuator to 6 dB, the desired signal amplitude at the DUT has to be within 0.25 V to 1.25 V. If you had set the attenuator to 20 dB, the legal amplitude voltage range would be 1/10th of the full amplitude, which means 0.05 V to 0.25 V.

Note that the recalculation assumes that a 50 Ω termination is used (which is mandatory).

4 Adjust the *Crossing Point*, if desired.

The vernier allows you to move the crossing point of the generated signal.

The following figure shows an example.

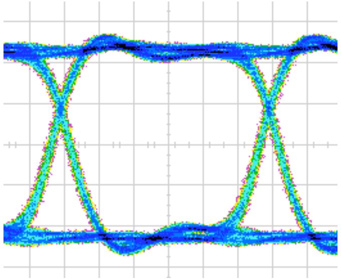


Figure 159 **Crossing Point above 50 %**

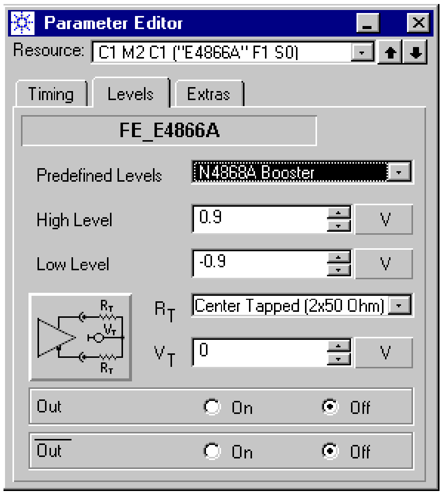
TIP If you wish to move quickly from one channel to the next, click the down/up arrows in the upper right-hand corner of the Parameter Editor window.

NOTE Besides the output parameters you must also adjust the input voltage(s).

The input voltages of the amplifiers are the output voltages of the connected 10.8 Gbit/s generators. They have to be set on the Levels page of the generators.

The Parameter Editor provides an own *Predefined Level* for the connection of E4866A 10.8 Gbit/s data generator modules to an N4868A booster module.

You need only choose the level “N4868A Booster” from the list.



Resource: C1 M2 C1 ("E4866A" F1 S0)

FE_E4866A

Predefined Levels: N4868A Booster

High Level: 0.9 V

Low Level: -0.9 V

R_T: Center Tapped (2x50 Ohm)

V_T: 0 V

Out ○ On ● Off

$\overline{\text{Out}}$ ○ On ● Off

Figure 160 Setting the Booster Module Input Voltage

Choosing the Kind of Measurement

Once the DUT has been modeled, its pins have been connected to instrument channels, and the channel parameters have been specified, it is time to tell the system what kind of measurement is going to be performed.

This has to be done before setting up the stream of generated and expected data, because different tests require different settings. Although the procedure for setting up the data sequence is always the same, the available segment options and result displays depend on the selected kind of measurement.

NOTE It is recommended to save settings repeatedly during test setup. You should take care of the chosen measurement type and save the setting under a file name that indicates the kind of measurement.

The kind of measurement is chosen from the Measurement Configuration window.

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial*.

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *“Choosing the Test Measurement Mode”*.

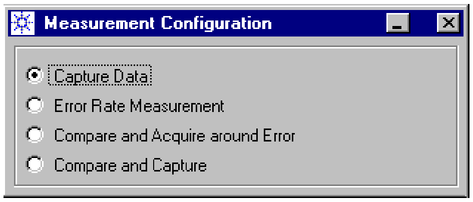
How to Open the Measurement Configuration Window

To open the Measurement Configuration window:

1 Click the Measurement Configuration icon in the tool bar.

Alternatively, you can also use the corresponding option of the *Go* menu.

The Measurement Configuration window appears.



Measurement Configuration

● Capture Data
○ Error Rate Measurement
○ Compare and Acquire around Error
○ Compare and Capture

Figure 161 Measurement Configuration Window

How to Set the Measurement Configuration

The Measurement Configuration window provides four options:

- Capture Data
- Error Rate Measurement
- Compare and Acquire Around Error
- Compare and Capture

Capture Data

In this measurement mode the system captures the data received by the analyzer connectors until either the test sequence is finished, or until the memory is full, or the Stop button was clicked.

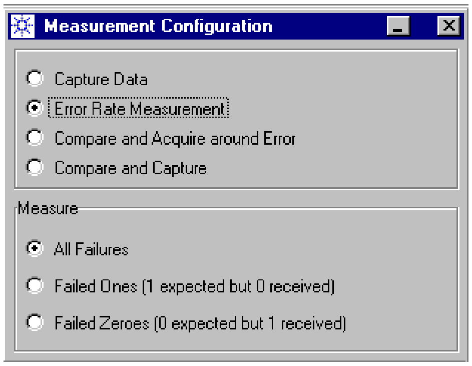
After the test has finished the recorded data can be viewed in the Error State Display and with the Waveform Viewer.

Error Rate Measurement

In this measurement mode the system continually samples incoming data applied to the analyzer connectors and compares the data in real time with expected data. The errors are counted and the bit error rate is calculated.

The result can be reviewed in the Bit Error Rate Display.

If you have chosen Error Rate Measurement, you have the following additional options:



Measurement Configuration

○ Capture Data
● Error Rate Measurement
○ Compare and Acquire around Error
○ Compare and Capture

Measure

● All Failures
○ Failed Ones (1 expected but 0 received)
○ Failed Zeroes (0 expected but 1 received)

Figure 162 Measurement Configuration Window for Error Rate Measurement

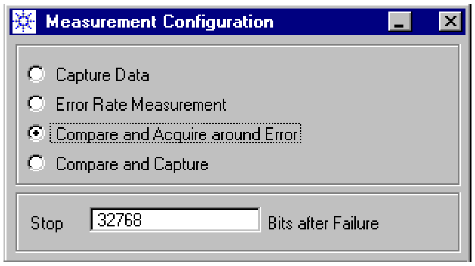
Select the kind of errors you wish to be counted.

Compare and Acquire Around Error

In this measurement mode the system records the data applied to the analyzer connectors until the test sequence is completed. The acquired data is compared in real time with expected data. As soon as an error occurs the system starts a counter.

The advantage of this mode is, that it is possible to define how long data should be recorded after the failure occurred, so that a pre- and post history around the error is captured and can be analyzed. The result can be reviewed in the Error State Display and with the Waveform Viewer.

If you have chosen Compare and Acquire around Error, you have one additional option:



Measurement Configuration

○ Capture Data
○ Error Rate Measurement
● Compare and Acquire around Error
○ Compare and Capture

Stop 32768 Bits after Failure

Figure 163 **Measurement Configuration Window for Capture Around Error Measurement**

You can set the number of bitstream vectors to be stored after the error occurred.

If an error occurs, the test will stop after the specified number of vectors has been acquired. The test will, of course, also stop if the test sequence expires before that number is reached.

Compare and Capture

In this measurement mode the system captures data applied to the analyzer connectors until the sequence is finished or the Stop button is pressed. While capturing, the system also compares the captured data with the expected data in real time.

The result can be reviewed in the Error State Display and with the Waveform Viewer.

Creating the Stream of Generated and Expected Data

After you have chosen the kind of measurement to be performed, you can build the test sequence.

The data to be generated or expected is embedded in a sequence. Three tools are provided for manipulating that sequence:

- Standard Mode Sequence Editor
- Detail Mode Sequence Editor
- Data/Sequence Editor

All the Sequence Editors can be started from the *Go* menu.

You can also click the Sequence Editor icon in the tool bar.

This opens the Standard Mode Sequence Editor, if the sequence conforms to the rules for a BER measurement. If not, the Detail Mode Sequence Editor is started.

NOTE If you are creating a sequence for an E4867A 10.8 Gbit/s data analyzer module, ensure that the sequence and test data conform to the special characteristics of that analyzer. See *"Special E4867A characteristics" on page 50*.

For details see:

- *"The Standard Mode Sequence Editor" on page 266*
- *"The Detail Mode Sequence Editor" on page 286*
- *"Using the Data/Sequence Editor" on page 335*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through:

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *“Creating a User-Defined Data Sequence”*.

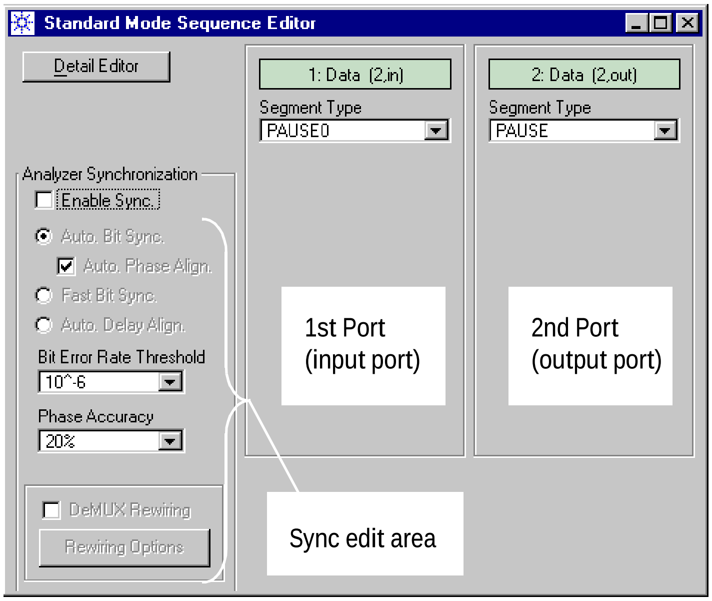
The Standard Mode Sequence Editor

The Standard Mode Sequence Editor is first of all meant for quick and easy setup of tests where generated and expected data are infinitely looped, such as bit error rate measurements.

For a simple and straightforward bit error rate measurement there is no need to worry about all the details of sequence blocks, loops, data segments, triggers, events, and so on.

You need only specify the PRBS polynomial or the data pattern to be used and are ready to run the test. Once it is started, the test will run until the Stop button is clicked.

If you have set up a new device, the Standard Mode Sequence Editor shows a window like the following:



1: Data (2,in)
Segment Type: PAUSE0

2: Data (2,out)
Segment Type: PAUSE

Analyzer Synchronization
☐ Enable Sync.
● Auto. Bit Sync.
☑ Auto. Phase Align.
○ Fast Bit Sync.
○ Auto. Delay Align.
Bit Error Rate Threshold: 10^-6
Phase Accuracy: 20%
☐ DeMUX Rewiring

Figure 164 Standard Mode Sequence Editor Window

The window has one panel for each DUT data port. The default segments are PAUSE0 for DUT input ports (connected to generator frontends), and PAUSE for output ports (connected to analyzers).

How to Use the Standard Mode Sequence Editor

You can:

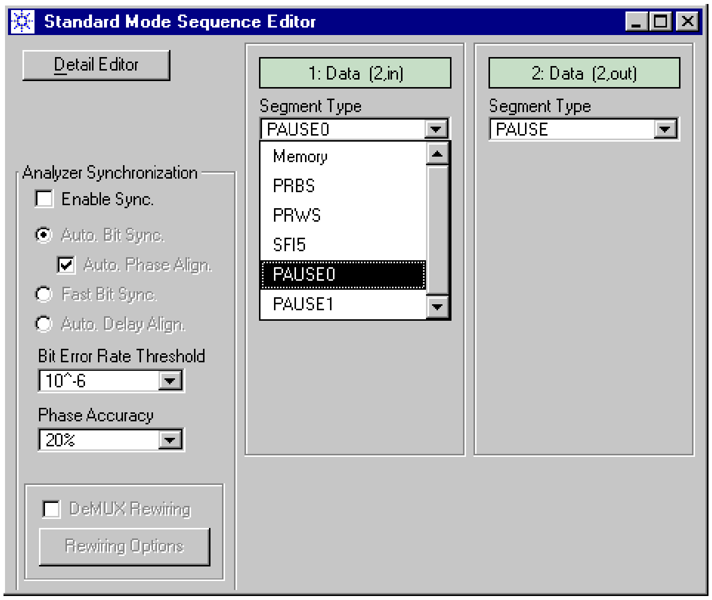
- Replace the current segments by new or existing segments
- Change segment properties
- Switch to the Detail Mode Sequence Editor
- Enable/Disable the automatic analyzer sampling point adjustment
- Edit the synchronization criteria

You cannot enable *Fast Bit Synchronization*. This is only possible when the Detail Mode Sequence Editor is active.

Replacing a Segment

To replace the current segment of a port by a different segment:

1 Open the *Segment Type* selection box.



Segment Type: PAUSE0

Segment Type: PAUSE

☐ Enable Sync.

● Auto. Bit Sync.

☑ Auto. Phase Align.

○ Fast Bit Sync.

○ Auto. Delay Align.

Bit Error Rate Threshold: 10^-6

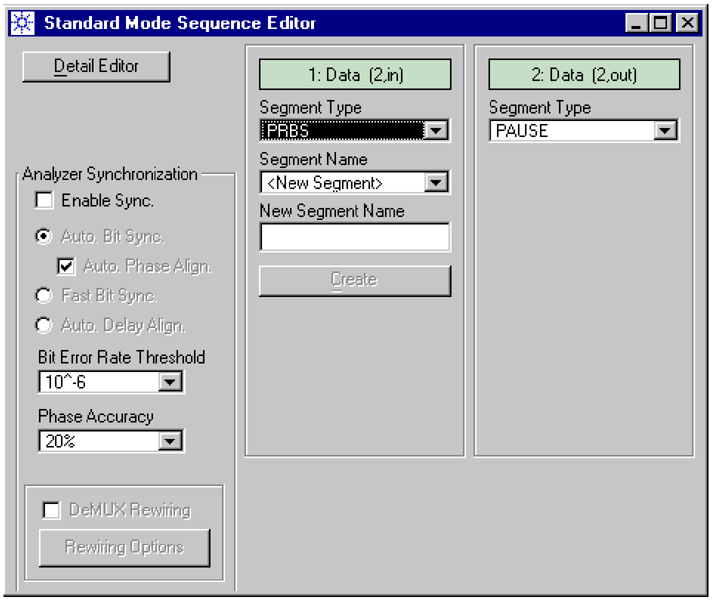
Phase Accuracy: 20%

☐ DeMUX Rewiring

Figure 165 Segment Type Selection

2 Choose the segment type: Memory-based, PRBS, or PRWS.

Note that *SFI5* is a special type of PRxS segment. It is meant for generating random data that is formatted according to SFI-5 standards. For details see the manual *Testing SFI-5 Devices*.



Standard Mode Sequence Editor

Analyzer Synchronization

□ Enable Sync.

● Auto. Bit Sync.

☑ Auto. Phase Align.

○ Fast Bit Sync.

○ Auto. Delay Align.

Bit Error Rate Threshold: 10^-6

Phase Accuracy: 20%

□ DeMUX Rewiring

1: Data (2,in)

Segment Type: PRBS

Segment Name: <New Segment>

New Segment Name:

2: Data (2,out)

Segment Type: PAUSE

Figure 166 Search Existing or Create New Segment

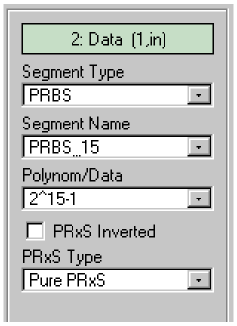
3 Choose a segment from the *Segment Name* browser, if you wish to use a segment that has been created previously

or

enter a *New Segment Name*, if you wish to create a new segment, and click *Create*.

Changing a Segment

If you have created a new PRBS/PRWS segment or chosen a PRBS/PRWS segment from the **local segment pool** (see also *"Data Segments" on page 81*), you can now change its polynomial and type.



2: Data (1,in)

Segment Type: PRBS

Segment Name: PRBS_15

Polynom/Data: 2^15-1

□ PRxS Inverted

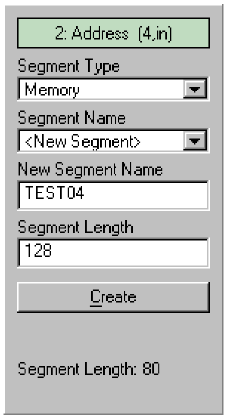
PRxS Type: Pure PRxS

Figure 167 **PRBS Parameters**

For a description of pure and non-pure PRBS/PRWS please refer to *"Appendix B: PRBS/PRWS Data Segments" on page 425.*

NOTE For 2^{23}–1 or 2^{31}–1 polynomials only *pure PRxS* is supported.

If you are creating a new memory-type segment, you can specify its length. The minimum segment length is equal to the length of the block and indicated on the panel:



2: Address (4,in)

Segment Type: Memory

Segment Name: <New Segment>

New Segment Name: TEST04

Segment Length: 128

Segment Length: 80

Figure 168 **Creating a New Memory Segment**

If you have created a new memory-type segment or chosen a memory segment from the local segment pool, you can now click *Edit* and change the contents of the segment.

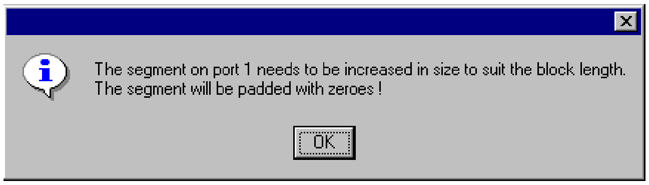
For details see *"How to Create a New Segment" on page 310.*

A new segment is automatically stored in the **local segment pool** which is associated with the current setting. Local segments can be directly edited with the Standard Mode Sequence Editor.

TIP Use the Segment Editor, if you have chosen a segment from the **global segment pool** and wish to inspect or change its contents. See also *"Data Segments" on page 81*.

Note that the size of a memory-type segment (length and width) may exceed but must not remain under the size of the block. Additional restrictions apply, if *Automatic Bit Synchronization* is enabled (see *"Block Length and Segment Length" on page 284*).

If the segment is too large, only a portion is used. If the segment is too small and is a local segment, its size is automatically increased. You get a message like the following:#



The segment on port 1 needs to be increased in size to suit the block length.
The segment will be padded with zeroes !

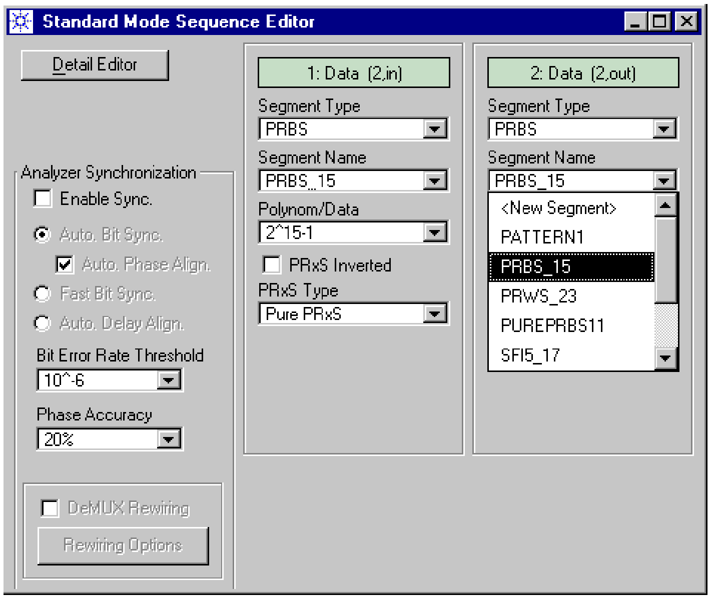
Figure 169 Segment-will-be-increased Message

To use the segment assigned to one port also for a different port:

1 Select the same *Segment Type*.

2 Choose the segment from the *Segment Name* browser.

The browser shows local as well as global segments.



Standard Mode Sequence Editor

Analyzer Synchronization
☐ Enable Sync.
● Auto. Bit Sync.
☑ Auto. Phase Align.
○ Fast Bit Sync.
○ Auto. Delay Align.
Bit Error Rate Threshold: 10^-6
Phase Accuracy: 20%
☐ DeMUX Rewiring

1: Data (2,in)
Segment Type: PRBS
Segment Name: PRBS_15
Polynom/Data: 2^15-1
☐ PRxS Inverted
PRxS Type: Pure PRxS

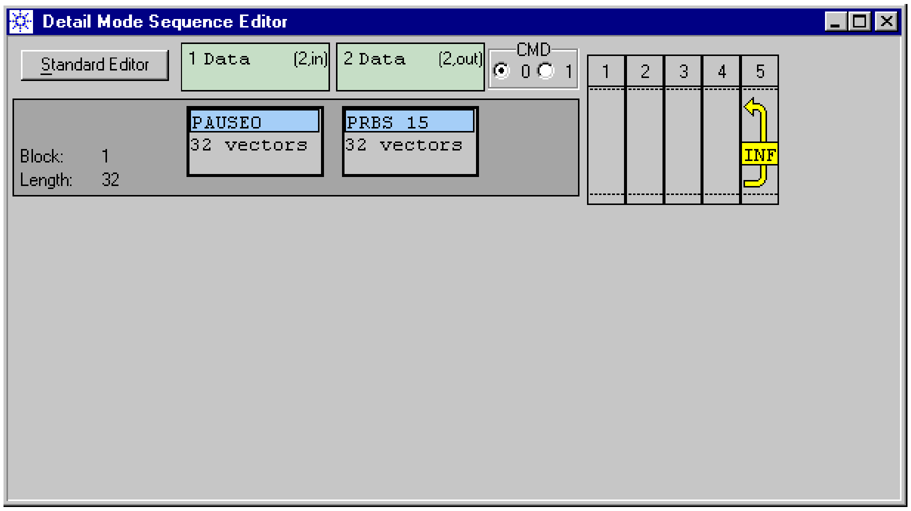
2: Data (2,out)
Segment Type: PRBS
Segment Name: PRBS_15

Figure 170 Segment Name Browser

Switching from Standard to Detail Mode Sequence Editor

You can always switch to the Detail Mode Sequence Editor by clicking the *Detail Editor* button.

The Detail Mode Sequence Editor gives you just another view:



Block: 1
Length: 32

PAUSE0
32 vectors

PRBS 15
32 vectors

Figure 171 Sequence in Detail View

It shows that the generated and expected data will be infinitely repeated.

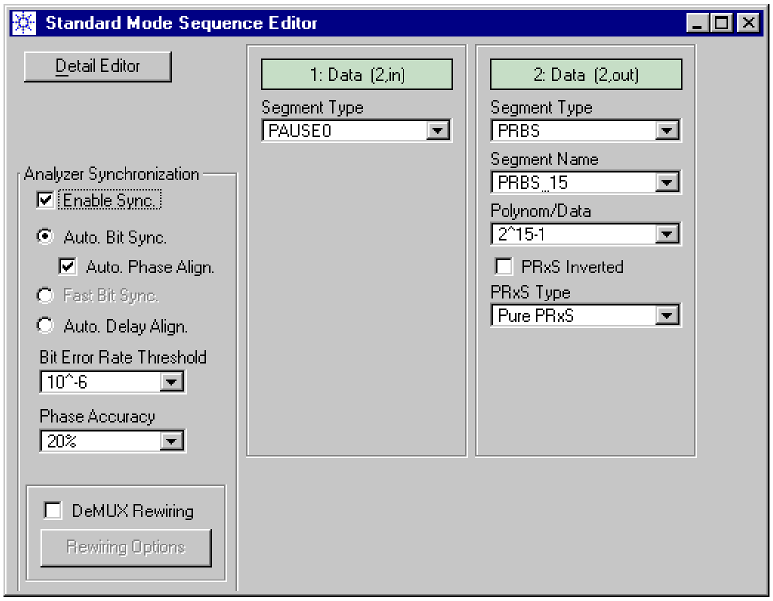
As long as you do not change the sequence with the Detail Mode Sequence Editor, you can always return to the Standard Mode Sequence Editor by clicking the Standard Editor button.

How to Synchronize an Analyzer With Incoming Data

The automatic analyzer sampling point adjustment with incoming data is available for both the Standard and the Detail Mode Sequence Editors.

For a detailed description of this feature please refer to *"Principles of Analyzer Sampling Point Adjustment" on page 87*.

Analyzer synchronization can be directly enabled/disabled and edited in the Standard Mode Sequence Editor window:



Analyzer Synchronization

☑ Enable Sync.

● Auto. Bit Sync.

☑ Auto. Phase Align.

○ Fast Bit Sync.

○ Auto. Delay Align.

Bit Error Rate Threshold: 10^-6

Phase Accuracy: 20%

□ DeMUX Rewiring

1: Data (2,in)

Segment Type: PAUSE0

2: Data (2,out)

Segment Type: PRBS

Segment Name: PRBS_15

Polynom/Data: 2^15-1

□ PRxS Inverted

PRxS Type: Pure PRxS

Figure 172 Synchronization Enabled in the Standard Mode Sequence Editor

NOTE You cannot enable *Fast Bit Synchronization* from the Standard Mode Sequence Editor. This is only possible when the Detail Mode Sequence Editor is active.

For the Detail Mode Sequence Editor, the *Enable/Disable Sync* function is an item found in the *Edit* menu and in the context menu of a block.

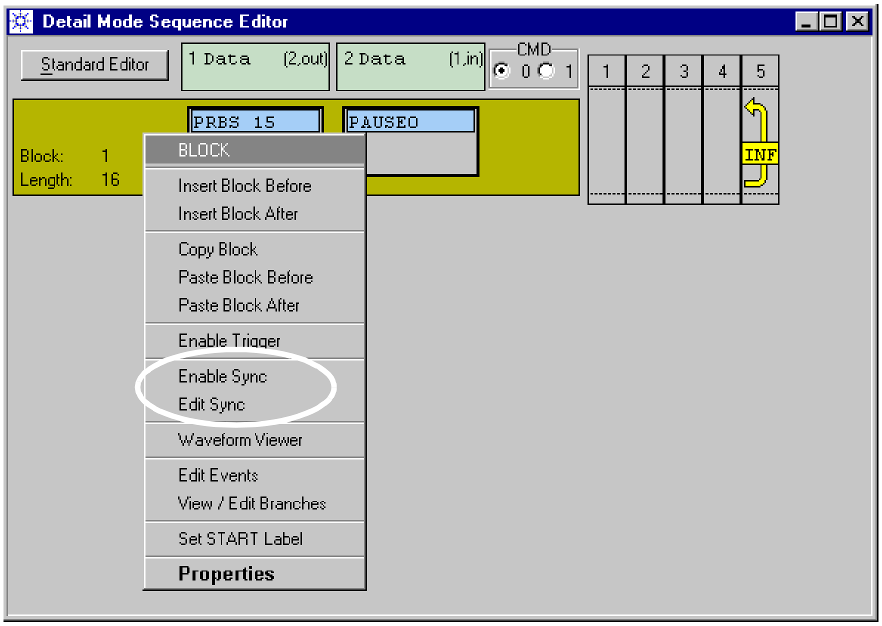


Figure 173 **Block Context Menu**

To use automatic analyzer synchronization:

1 Enable the checkbox

or

from the menu, choose *Enable Sync*.

If you are using the Standard Mode Sequence Editor, a synchronization block is automatically inserted into the sequence. This block is placed ahead of the infinite loop. It has the same length and contains the same data as the test block.

If you are using the Detail Mode Sequence Editor, the currently highlighted block becomes the synchronization block. This should be the first block of the sequence that contains expected data.

When the test is run, the synchronization block is automatically repeated until the synchronization criteria are met. Then the sequencer continues with the next block.

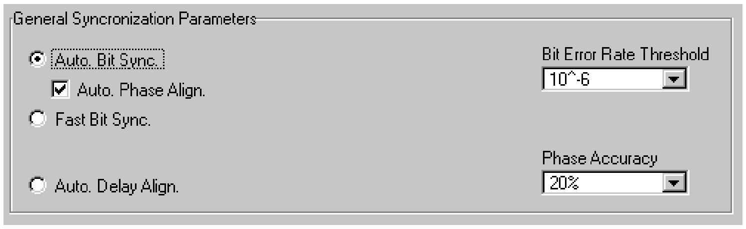
Fast Bit Synchronization behaves differently. For details see below.

2 If the automatic analyzer synchronization is enabled, you can click *Disable Sync* to deactivate it.

NOTE If you disable the automatic analyzer sampling point adjustment from the Standard Mode Sequence Editor, the synchronization block is removed from the sequence. Only the test block with the infinite loop remains.

If you disable the automatic analyzer sampling point adjustment from the Detail Mode Sequence Editor, only the sync label is removed from the block.

3 If you have opened the Detail Mode Sequence Editor, choose *Edit Sync* from the *Edit* menu or the context menu of the first block. This provides the access to the synchronization parameters that are directly accessible in the Standard Mode Sequence Editor.



General Syncronization Parameters

● Auto. Bit Sync.

☑ Auto. Phase Align.

○ Fast Bit Sync.

○ Auto. Delay Align.

Bit Error Rate Threshold: 10^-6

Phase Accuracy: 20%

Figure 174 General Synchronization Parameters

4 Select the synchronization method you wish to use. Choices are *Automatic Bit Synchronization* or *Automatic Delay Alignment*. When the Detail Mode Sequence Editor is active, you can also choose *Fast Bit Synchronization*.

At this point, a few explanations may be helpful. For details please refer to *"Principles of Analyzer Sampling Point Adjustment" on page 87*.

Automatic Bit Synchronization

Automatic Bit Synchronization has the option to enable or disable *Automatic Phase Alignment*:

Automatic Bit Synchronization without Automatic Phase Alignment is used if the total delay from test start is unknown but a certain edge delay relative to the analyzer clock is expected.

Automatic Bit Synchronization with Automatic Phase Alignment is used if the delay is completely unknown.

PRBS data may be sent and expected. Memory-based data may also be expected by a pure analyzing system.

The final delay status is indicated by the Parameter Editor.

Automatic Bit Synchronization does not report the number of clock periods that have passed since test start, but only the phase shift relative to the clock.

Auto Bit Sync without Auto Phase Alignment

If Automatic Phase Alignment is disabled, then the analyzer uses the start delay that has been specified with the Parameter Editor to determine the sampling point delay relative to its clock. It then samples the incoming data until the incoming data matches the expected pattern with an adequate accuracy.

Once this accuracy is reached, then the incoming bits are aligned with the expected bits—the analyzer is synchronized with the incoming data.

Auto Bit Sync with Auto Phase Alignment

If Automatic Phase Alignment is enabled, then the analyzer fully automatically adjusts itself to capture the incoming data at the optimum sampling point.

It shifts the sampling point stepwise in both directions until the specified bit error rate is reached. The width of these steps is adjustable. The analyzer then measures the width of the eye diagram and positions the sampling point at the optimum.

Automatic Delay Alignment

Automatic Delay Alignment is used if the delay between the start of the test and the incoming data is coarsely known and set as the start delay with the Parameter Editor. PRBS as well as memory-type data can be generated and expected.

The analyzer starts after the start delay has elapsed. It then shifts the sampling point within a certain range (±50 ns for an E4832A data generator/analyzer module and ±10 ns for an E4861A module) until it recognizes the expected pattern with an adequate accuracy.

After that, Automatic Delay Alignment measures the width of the eye diagram and positions the sampling point in the middle. The final delay status is indicated by the Parameter Editor.

Automatic Delay Alignment reports the full delay since test start.

Fast Bit Synchronization

Fast Bit Synchronization aligns received and expected data. It can only be used with pure PRBS/PRWS data. It does not change the analyzer sampling point. The analyzer delay has to be determined and set beforehand.

For a description of Fast Bit Synchronization please refer to *“Fast Bit Synchronization” on page 97*.

Sync block indicators

The Detail Mode Sequence Editor uses the following symbols to highlight the synchronization block:

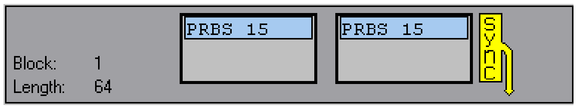


Figure 175 Auto Bit Sync Block Indicator

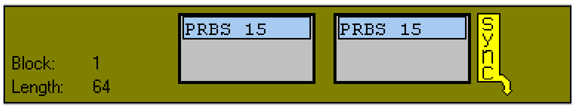


Figure 176 Auto Delay Alignment Block Indicator

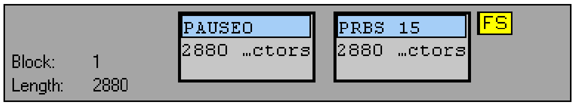


Figure 177 Fast Bit Sync Block Indicator

The symbols allow to identify the synchronization method that has been chosen.

If you have chosen Fast Bit Synchronization, you are done. If not:

5 Set the *Bit Error Rate Threshold*.

This threshold defines the “adequate accuracy” which depends on the DUT and the test requirements. The synchronization process is not complete unless the actual BER remains under this threshold.

6 Set the *Phase Accuracy*.

The phase accuracy can be set between 20 % and 1 %. This defines the number of steps performed by the phase optimizing algorithm (5 to 100) and has an impact on the speed of the synchronization process.

NOTE Once the test is started, the success or failure of the automatic delay adjustment can be seen in the BER display window: A terminal on which the connected analyzer could not synchronize shows no bit error rate at all.

7 Decide on using DEMUX Rewiring.

How to Specify DEMUX Rewiring Parameters

DEMUX rewiring is a special feature for synchronizing on the output of a demultiplexer. Many demultiplexers have the peculiarity that one cannot predict at which of the output terminals the first bit of the serial input bit stream appears.

DEMUX rewiring should always be used when testing a demultiplexer with memory-based data. DEMUX rewiring rearranges the order of the demultiplexer terminals so that the bits arriving at the analyzers can be compared with the data stored in the segment. This comparison, in turn, is necessary for both automatic delay adjustment and the test.

The principles of DEMUX rewiring are explained in *"Automatic Rewiring of Demultiplexer Terminals" on page 119*.

NOTE DEMUX rewiring has to be used at a ParBERT 43G error detector system whenever the test data is not pure PRBS/PRWS.

If you have enabled DEMUX Rewiring, you must enter the DEMUX rewiring parameters.

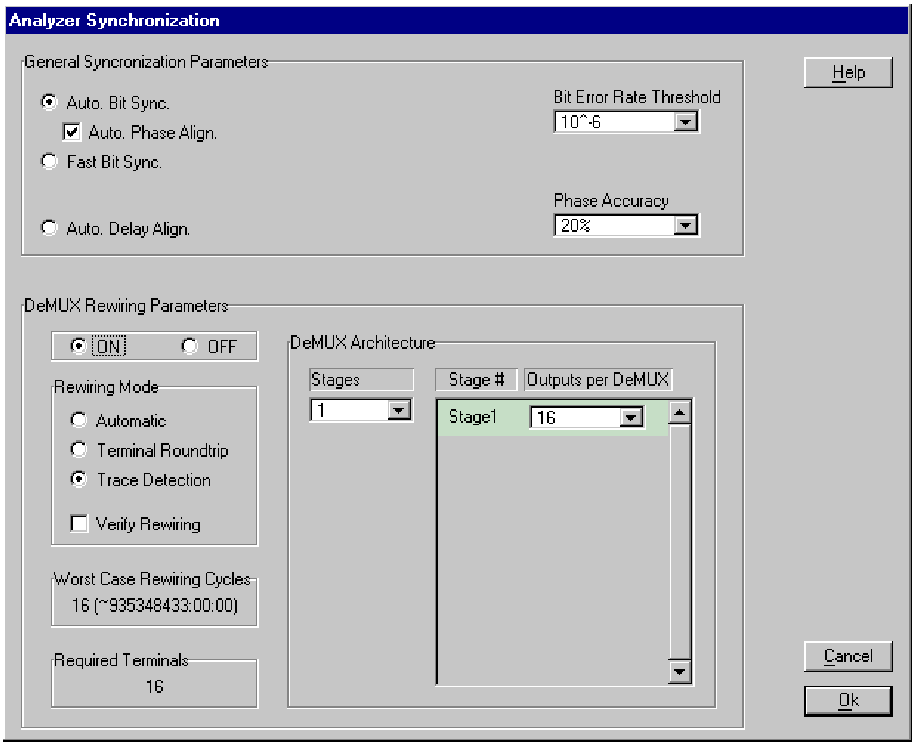
To set the DEMUX rewiring parameters:

1 In the Standard Mode Sequence Editor, click the Rewiring Options button

  or

  in the Detail Mode Sequence Editor, choose *Edit Sync* from the *Edit* menu or the context menu of the synchronization block.

This opens the following window:



Analyzer Synchronization

General Syncronization Parameters

● Auto. Bit Sync.

☑ Auto. Phase Align.

○ Fast Bit Sync.

○ Auto. Delay Align.

Bit Error Rate Threshold: 10^-6

Phase Accuracy: 20%

DeMUX Rewiring Parameters

● ON ○ OFF

Rewiring Mode

○ Automatic

○ Terminal Roundtrip

● Trace Detection

□ Verify Rewiring

Worst Case Rewiring Cycles: 16 (~935348433:00:00)

Required Terminals: 16

DeMUX Architecture

Stages: 1

Stage #	Outputs per DeMUX
Stage1	16

Figure 178 Automatic Analyzer Delay Adjustment Control Window

The default settings shown in the figure above are adequate to a ParBERT 43G error detector system using the E4869A DEMUX module.

2 Choose the rewiring mode. Choices are:

- *Automatic*—the automatic mode tries first to rewire with the trace detection method and then, if that failed, with the terminal roundtrip method.
- *Terminal Roundtrip*—tests all possible permutations and takes some time, especially for a multi-stage demultiplexer.
- *Trace Detection*—a speed-optimized approach.

  Trace Detection requires that the first 48 expected bits of each terminal are unequivocal and unique within the segment used for the synchronization. This is the same precondition that applies to Automatic Bit Synchronization with memory data.

3 Enable or disable *Verify Rewiring*.

If *Verify Rewiring* is enabled, the system informs you about problems that might arise.

4 Enter the DEMUX architecture.

The figure above refers to a simple, one-stage 1:16 demultiplexer. A multi-stage demultiplexer may have an architecture as illustrated below:

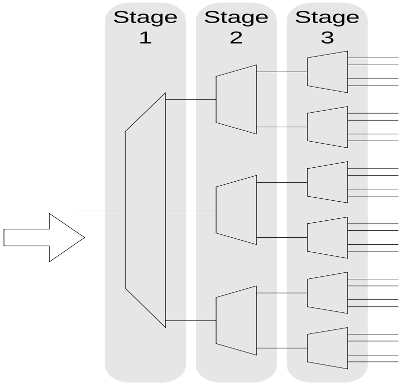
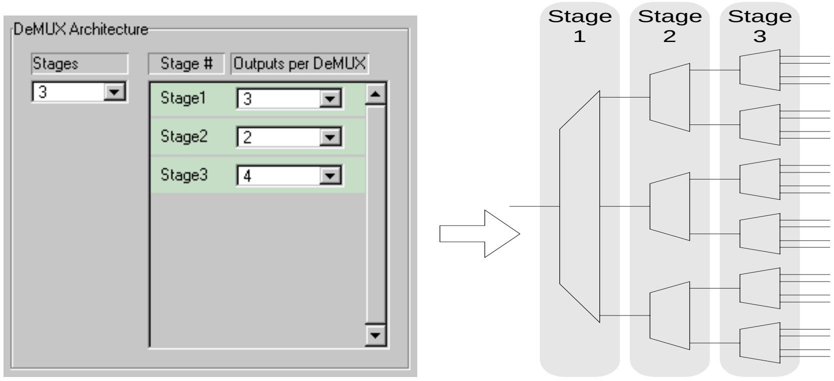


Figure 179 Three-Stage Demultiplexer

On stage one, the number of outputs per demultiplexer is three. On stage two, the number of outputs per demultiplexer is two. On stage three, the number of outputs is four.

This architecture is entered as shown in the following figure:



DeMUX Architecture

Stages: 3

Stage #	Outputs per DeMUX
Stage1	3
Stage2	2
Stage3	4

Figure 180 Specification of a Three-Stage Demultiplexer

Only symmetrical architectures are supported. This means that all demultiplexers of one stage have to be identical.

5 Check the number of *Required Terminals*.

This number (24 in the example above) is automatically calculated. All these terminals have to be connected to analyzer frontends.

6 Check the number of *Worst Case Rewiring Cycles* and the expected time.

The number of *Stages* as well as the number of *Outputs per DeMUX* have an impact on the number of possible permutations and hence on the duration of the rewiring and synchronization process.

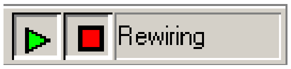
The expected duration that is also displayed is an estimation, assuming a synchronization segment of moderate length. The actual duration may differ.

For the *trace detection* algorithm, the number of rewiring cycles for a multi-stage demultiplexer is calculated as *number of terminals* + *1*.

For the *terminal roundtrip* algorithm, all possible permutations are taken into account. Every demultiplexer of every stage may exhibit that unpredictable behavior. Depending on the demultiplexer architecture, terminal roundtrip may take half a minute or even hours.

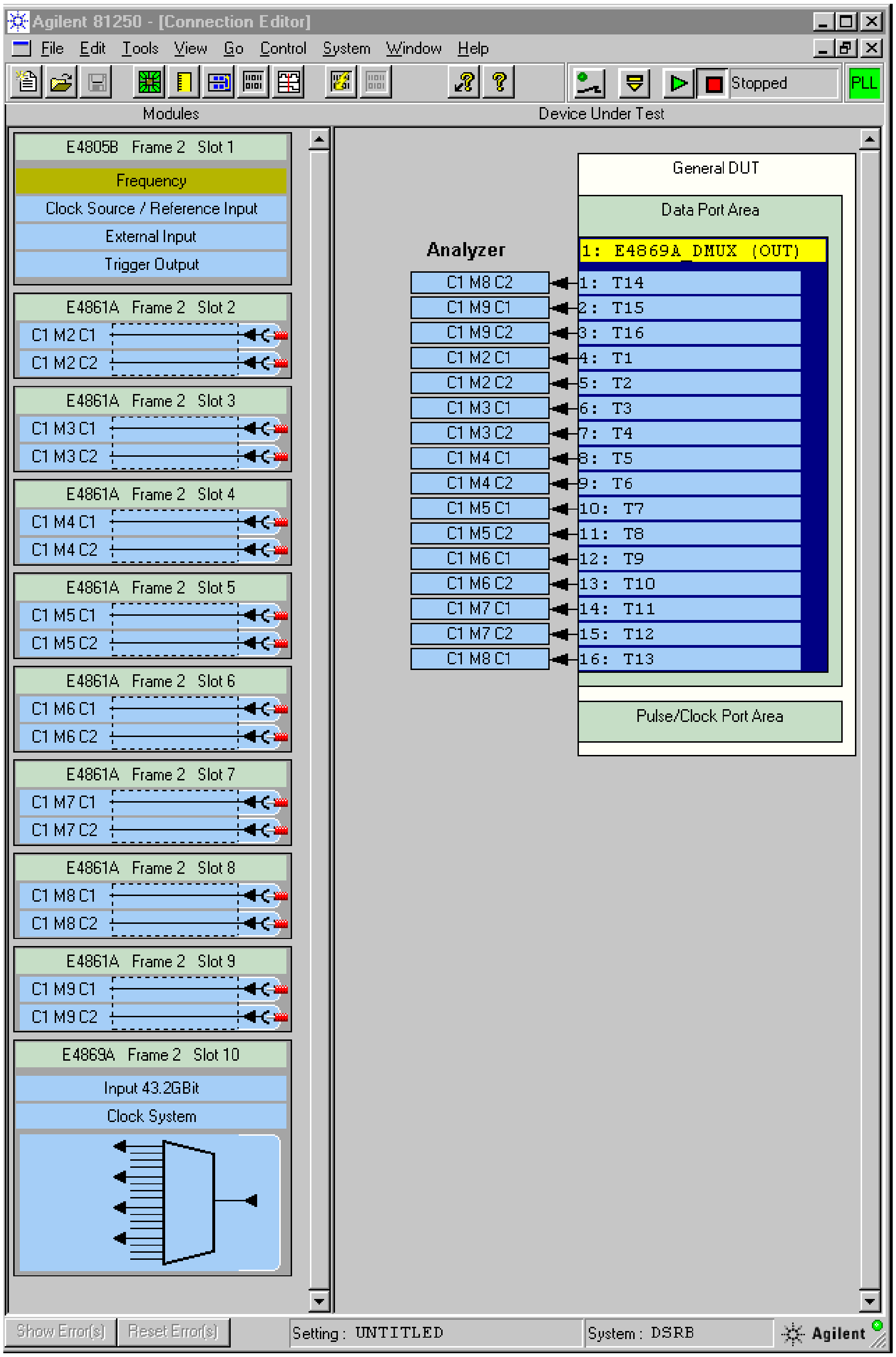
“Overflow” is indicated, if the estimated time exceeds 24 hours. “Missing Parameters” is indicated, if the demultiplexer is not fully described. You cannot leave the dialog until such errors have been corrected.

NOTE Once the test is started, it begins with DEMUX rewiring and automatic analyzer delay adjustment. During this phase, the user interface does not respond.



The rewiring phase is indicated by the side of the Run/Stop buttons.

After DEMUX rewiring, the Connection Editor displays the new terminal order. For an E4869A DEMUX module, this may look as shown in the following figure:



Analyzer	1: E4869A_DMUX (OUT)
C1 M8 C2	1: T14
C1 M9 C1	2: T15
C1 M9 C2	3: T16
C1 M2 C1	4: T1
C1 M2 C2	5: T2
C1 M3 C1	6: T3
C1 M3 C2	7: T4
C1 M4 C1	8: T5
C1 M4 C2	9: T6
C1 M5 C1	10: T7
C1 M5 C2	11: T8
C1 M6 C1	12: T9
C1 M6 C2	13: T10
C1 M7 C1	14: T11
C1 M7 C2	15: T12
C1 M8 C1	16: T13

Setting : UNTITLED

System : DSRB

Figure 181 DEMUX Rewiring Result

Note that compared to the original, automatic configuration, only the order of the terminals has changed, not the connections. Terminal T14 is now at the first place. This corresponds to the first trace of the data segment.

So, DEMUX rewiring ensures that the incoming data can be compared with the expected, no matter on which terminal the first bit arrives.

Characteristics of the Standard Mode Sequence Editor

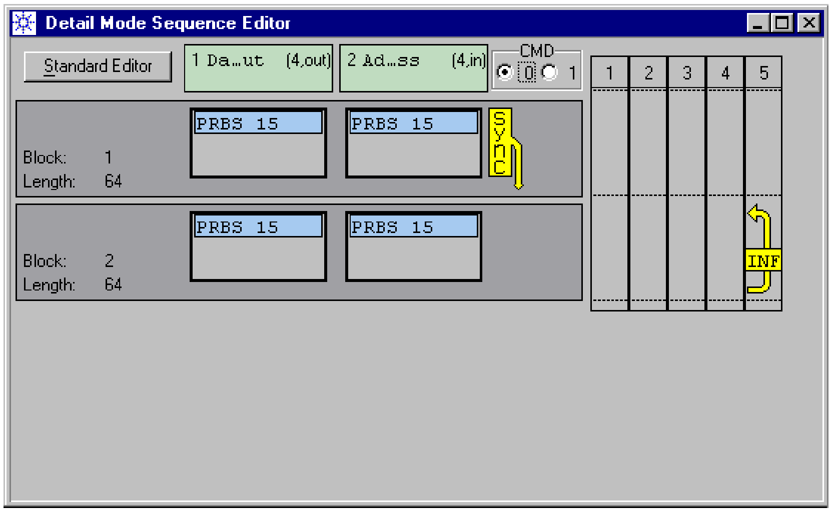
If a test sequence was edited with the Detail Mode Sequence Editor or the Data/Sequence Editor, it can happen that the Standard Mode Sequence Editor cannot be opened from the *Go* menu.

The reason is that the sequence does not conform to the rules for a simple bit error rate measurement.

BER Measurement Sequences

A BER measurement requires one single test block which is infinitely repeated. If the automatic analyzer sampling point adjustment is enabled, this block is preceded by a synchronization block.

A valid sequence with Automatic Bit Synchronization may look like this:



Block: 1
Length: 64

Block: 2
Length: 64

Figure 182 Test Sequence for BER Measurement With Bit Synchronization Enabled

The Standard Mode Sequence Editor can be used, if the test sequence meets the following criteria:

- The sequence contains one test block which is perpetually repeated.
- Different data segments may be used for different ports.

- The test block may be preceded by a synchronization block (a block with sync enabled and without loop). This block must be an unchanged copy of the test block (same size, same segments).
- The sequence does not process any internal or external events.
- A trigger (activated with the Detail Mode Sequence Editor) may be generated but is not shown in the Standard Mode Sequence Editor.
- The blocks have a certain length (which implies that the referenced segments must also have a corresponding minimum length).

Block Length and Segment Length

The Standard Mode Sequence Editor calculates the **block length** automatically. The block length has to be an integer multiple of the individual segment segment resolution of the port or channel (see *"Frequency Multiplier and Segment Resolution" on page 68*).

The **minimum length of a segment** is the individual segment resolution of the port or channel. This is the product of the general segment resolution times the FM factor of the port.

Example:
If the general segment resolution is 4 and the FM factor of the port is 2, then the minimum segment length is 8.

If a segment with a user-defined length is entered into a port with the Standard Mode Sequence Editor, its length is automatically adjusted to the next integer multiple of general SR × FM.

Example:
If you enter a segment with a length of 120 vectors and the minimum segment length is 16, then the segment length will be increased to 128.

NOTE Two special conditions apply to segments that specify expected data in conjunction with **Automatic Bit Synchronization** (not Auto Delay Alignment). If Automatic Bit Synchronization is enabled, then please observe:

- If a memory-based data segment is used for synchronization, the block length has to be a multiple of 32 × SR and the length of the segment has to be at least 32 × SR × FM.

  If the SFI-5 Post Processing Tool is used for analyzing SFI-5 data, the block length must additionally be a multiple of 1088. For details see the manual *Testing SFI-5 Devices*.

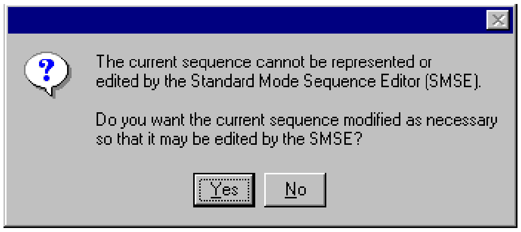
- If a non-pure PRBS-type segment is used for synchronization, the block length has to be exactly SR × PL (where PL is the length of the polynomial—for example 2^{13}–1 means 8191).

All this is automatically considered by the Standard Mode Sequence Editor, but not by the Detail Mode Sequence Editor which allows all kinds of modifications.

The Standard Mode Sequence Editor automatically sets the correct segment length and adjusts the block length if necessary. This is especially important, if some of the ports are operated at different clock rates.

Switching From Detail to Standard Mode Sequence Editor

If you try to switch from the Detail Mode Sequence Editor to the Standard Mode Sequence Editor, the following question may appear:



The current sequence cannot be represented or edited by the Standard Mode Sequence Editor (SMSE).

Do you want the current sequence modified as necessary so that it may be edited by the SMSE?

Figure 183 **Transition From Detail to Standard Mode Sequence Editor**

CAUTION

Click *No*, if you have created a sophisticated sequence.

Remember that the Standard Mode Sequence Editor is first of all meant for setting up a simple BER test, based on one block which is infinitely looped.

If the above question appears and you really wish to use the Standard Mode Sequence Editor, it may be a good idea to first save the current setting. If you have inadvertently clicked *Yes*, the only way to restore the previous sequence is to re-load the last setting.

If you click *Yes*, the following happens:

- All blocks of the sequence except the first one will be deleted.
- All events specified for the sequence are deleted.
- All loops are deleted.
- The remaining block is considered to be the test block. It gets an infinite loop.
- If automatic analyzer sampling point adjustment is enabled for the remaining block, the block is duplicated. The copy is inserted as block one and serves as a synchronization block (a block with sync enabled and without loop).

The Detail Mode Sequence Editor

The Detail Mode Sequence Editor allows to create and maintain individual test sequences.

A sequence consists of blocks. The blocks can be executed one after the other. Blocks and groups of blocks can also be repeated a specified number of times before the sequence continues. In addition, an endless loop can be specified – with the result that the sequence never ends.

If you activate the event recognition feature built into the system, the order of the block execution becomes variable. Based on specified events, you can leave a loop or even a block and continue with another block of the sequence.

The blocks reference data segments. These segments specify the generated and expected data patterns.

For details see:

"Contents of the Detail Mode Sequence Editor Window" on page 287

"How to Add, Move or Delete Blocks" on page 288

"How to Change Block Properties" on page 289

"How to Replace the Current Segment" on page 292

"How to Create and Change Loops" on page 296

"How to Specify Events and Reactions Upon Events" on page 298

"How to Synchronize an Analyzer With Incoming Data" on page 272

TIP After the sequence has been set up, it can also be inspected in detail and changed with the Data/Sequence Editor (see *"Using the Data/Sequence Editor" on page 335*).

Contents of the Detail Mode Sequence Editor Window

For a new device, the Detail Mode Sequence Editor shows one single block:

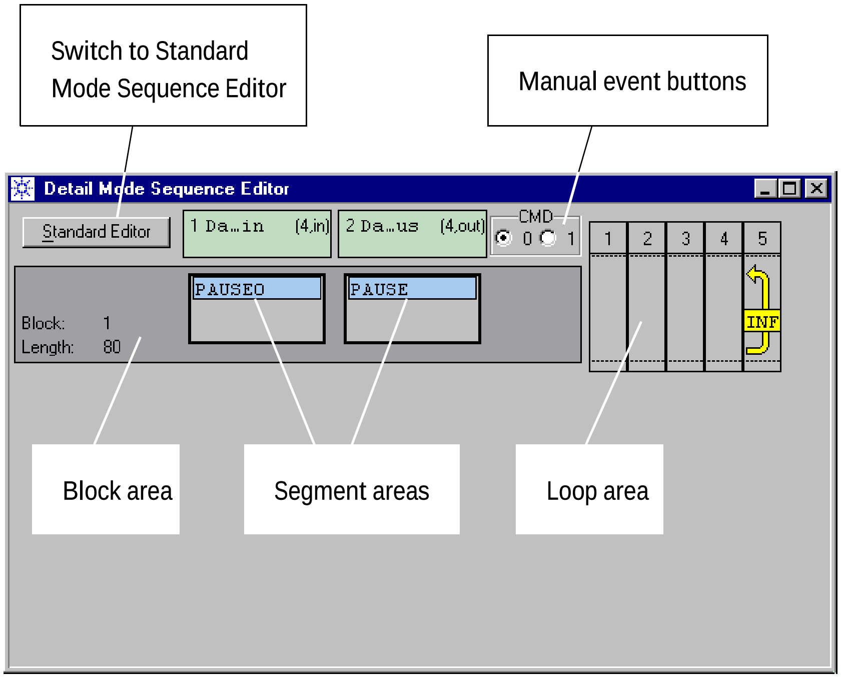


Figure 184 Default Sequence for a New Device With Two Data Ports

The block has a default length and includes default segments for all data input and data output ports that have been configured with the Connection Editor.

If the DUT setup includes more ports than can be shown, then a horizontal scroll bar is displayed to move to the hidden ports.

NOTE The default segments as well as the available pseudo segments depend on the type of port (DUT input or output) and on the kind of measurement.

At the right-hand side of the block is the loop area where repetitions can be specified.

How to Add, Move or Delete Blocks

A sequence often consists of more than one block. A sequence can contain up to 60 blocks. This number decreases if counted loops are used (see *"Hardware Dependencies" on page 85*).

You may wish to add blocks, delete blocks, or move blocks.

To manipulate the overall sequence:

1 Open the context menu of a block (right-click into the block area).

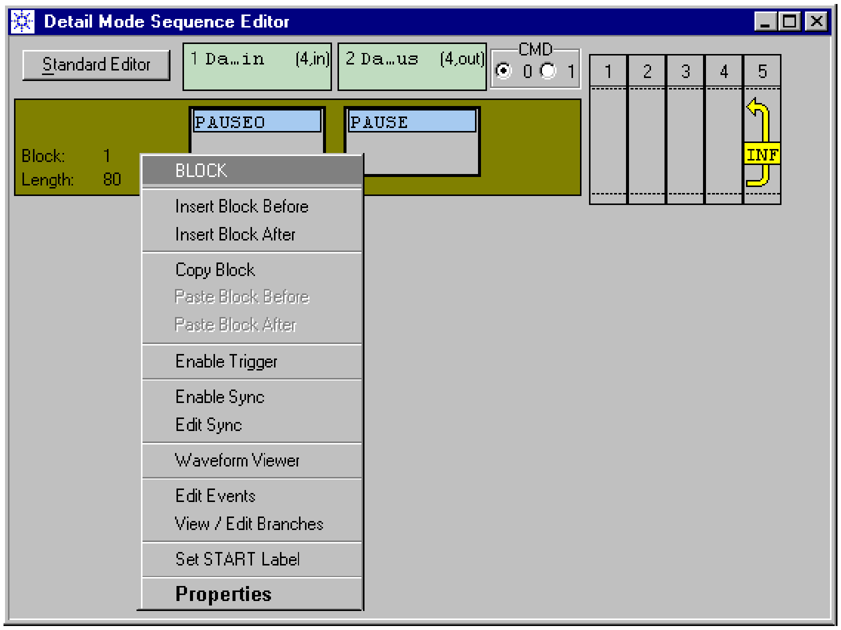


Figure 185 Block Menu

2 Choose the desired action:

- *Insert Block Before*: Inserts a new default block with default segments before the current block.
- *Insert Block After*: Inserts a new default block with default segments below the current block.
- *Copy Block*: Copies the chosen block to the clipboard.
- *Cut Block*: Copies the chosen block to the clipboard and removes it from the sequence (not available if the sequence contains only one block).
- *Paste Block Before*: Available, after a block has been copied or cut. Inserts the block from the clipboard above the current block.
- *Paste Block After*: Available, after a block has been copied or cut. Inserts the block from the clipboard below the current block.

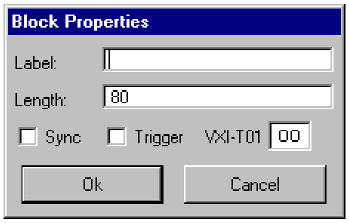
How to Change Block Properties

Block characteristics include block length, block label, and trigger output.

To change the block characteristics:

1 Double-click on the block area.

Alternatively, you can also open the context menu of the block and select *Properties*.



Label:

Length: 80

□ Sync □ Trigger VXI-T01 00

Figure 186 Block Properties Window

2 Enter appropriate data:

- *Label*: The block label should indicate the contents or purpose of the block. If the event recognition feature is used, the block label identifies the block that can be jumped to.

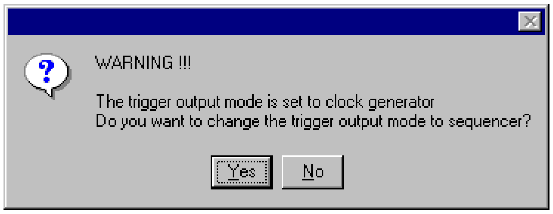
NOTE There are two block labels which have a special meaning:

The label START denotes the first block of a sequence. If this label is present, blocks above may exist, but are not processed when the test is run. The START label can also be assigned from the Block context menu by clicking *Set START label*.

The label END denotes the end of a sequence. This is an implicit label that should not be entered. The END label is used by the event recognition for terminating the test upon an event.

- *Length*: The length of the block must be a multiple of the segment resolution. For details see *"Frequency Multiplier and Segment Resolution" on page 68* and *"Block Length and Segment Length" on page 284*.
- *Sync*: The *Sync* enable button can be used to enable or disable automatic analyzer synchronization (see *"How to Use a Block for Analyzer Sampling Point Adjustment" on page 291*).
- *Trigger*: If you activate Trigger, a trigger is generated at the TRIGGER OUTPUT connector of the master clock module each time the block is started.

When you set the trigger, you may get the following message:



WARNING !!!

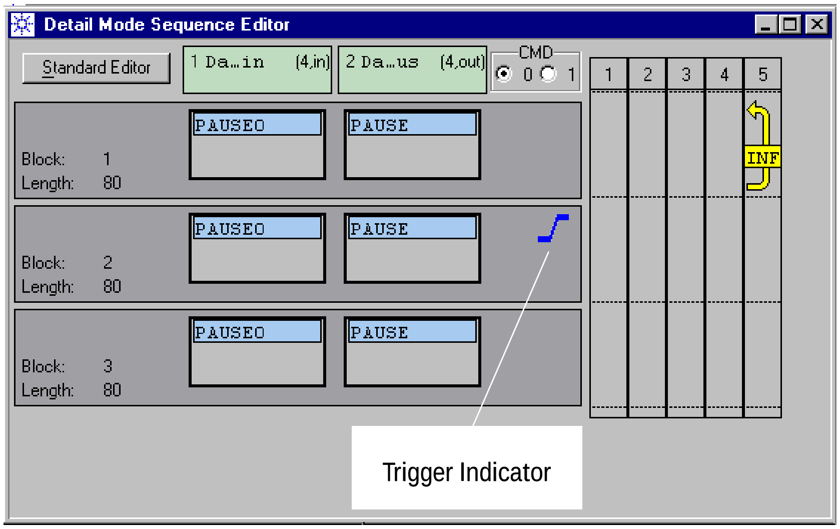
The trigger output mode is set to clock generator
Do you want to change the trigger output mode to sequencer?

Figure 187 Wrong-Trigger-Mode Warning

By default, the TRIGGER OUTPUT of the master clock module is set up as a clock generator which generates a continuous clock pulse. If you wish to generate single trigger pulses, click *Yes*.

See also *"How to Set the Characteristics of the Trigger Output" on page 192*.

The block area of the Sequence Editor indicates, whether a trigger is set.



Block: 1
Length: 80

Block: 2
Length: 80

Block: 3
Length: 80

Trigger Indicator

Figure 188 Block Trigger Indicator

– *VXI-T01*: You can also specify the setting of the VXI trigger lines T0/T1 at the beginning of the block.

  If you don‘t wish to change their status, enter xx.

  If you wish to activate the VXI triggers for controlling other VXI modules, ensure that you have not defined an event based on the status of the VXI trigger lines. See also *"How to Define Events" on page 303*.

How to Use a Block for Analyzer Sampling Point Adjustment

In principle, any block of the sequence can be used for synchronizing the analyzer channels with incoming data. But if automatic analyzer sampling point adjustment is required, measurements before synchronization usually don’t make much sense.

Meaningful options are:

- Place the synchronization block at the beginning of the sequence.
- If a delay is needed before synchronization (for example to allow a PLL to settle), ensure that the synchronization block is only preceded by Pause blocks.
- If you wish to keep an existing sequence, assign the START label to start the sequence execution at the synchronization block.

To create a synchronization block:

1 Select the block and open the block’s context menu.

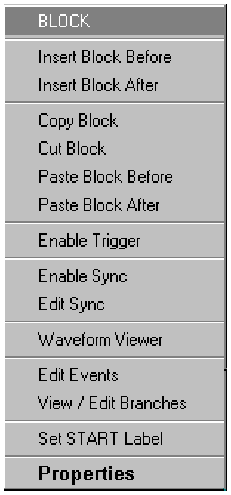


Figure 189 Block Context Menu

2 If it is not the first block, click *Set START Label* (not mandatory, but recommended).

   The execution of the sequence will start with this block.

3 Click *Enable Sync*.

4 Click *Edit Sync* to check and eventually change the synchronization criteria. For details see *“How to Synchronize an Analyzer With Incoming Data” on page 272.*

NOTE Especially if you wish to use Automatic Bit Synchronization, you may need special segments and may have to adjust the length of the block and the segments. See *"Block Length and Segment Length" on page 284*.

How to Replace the Current Segment

The segments contained in a block describe the data to be generated or expected.

The default segments that appear in new blocks are pseudo segments. They depend on the type of port (DUT input or output) and on the chosen kind of measurement. They can be replaced by a different pseudo segment or a real segment.

To change a segment:

1 Open the segment's context menu (right-click on the segment name).

You get a menu like the one shown below:

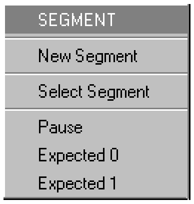


Figure 190 **Segment Selection Menu of the Sequence Editor**

2 Choose one of the options.

You can create a new segment, select an existing segment, or choose one of the available pseudo segments.

How to Replace a Segment by a New Real Segment

To replace the current segment by a new real segment:

1 Choose a segment and open the segment's context menu.

2 Select *New Segment*.

For details see *"How to Create a New Segment" on page 310*.

How to Replace a Segment by a Stored Segment

PRBS and memory segments that have been previously created are stored in the global or local segment pool.

To select a stored segment:

1 Select a segment and open the segment's context menu.

2 Choose *Select Segment*.

The segment selection window appears.

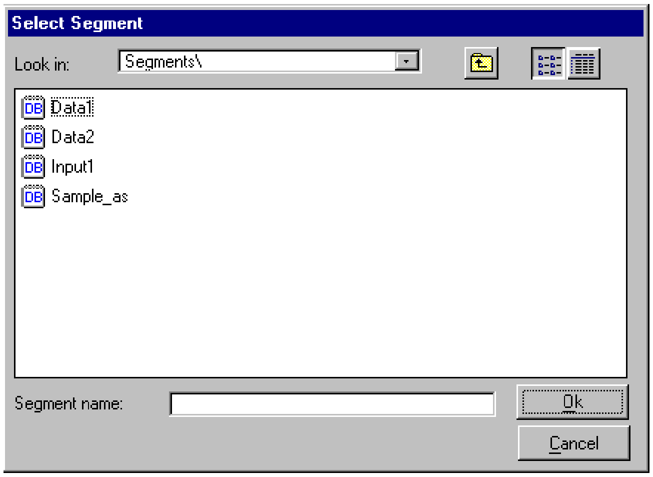


Figure 191 Select Segment Window

Per default, the window shows all accessible segments. You can change the directory to view only the global or local segment pool. Segments in the global segment pool can be accessed from all settings. Segments in the local segment pool can only be accessed from the current setting.

3 Select the segment you wish to insert into the block.

NOTE

If you intend to use a stored segment, please note: Not every segment fits to every block.

If the length and/or width of a segment is smaller than the block length or port width, then an error message is displayed when the sequence is downloaded. It is necessary to edit the segment to match the block length and port width.

If the length or width of a segment is larger than the blocklength or port width, then only a portion of the segment will be generated or expected. This portion starts from trace 0 and vector 0 of the segment. All traces which exceed the width of the port and all vectors which exceed the length of the block are ignored.

The lengths of the blocks have to be a multiple of the segment resolution which is a trade-off between the required system clock rate and the desired memory depth. For details see *"Frequency Multiplier and Segment Resolution" on page 68*.

See also *"Data Memory Usage" on page 82*, *"Segment Type Combinations" on page 83*, and *"How to Set the General System Frequency" on page 178*.

Additional restrictions apply for synchronization blocks if Automatic Bit Synchronization is used. See *"Block Length and Segment Length" on page 284*.

4 Confirm.

How to Replace a Segment by a Pseudo Segment

To select a pseudo segment in the Detail Mode Sequence Editor:

1 Select a segment and open the segment's context menu.

The lower part of the menu lists the available pseudo segments.

The available pseudo segments depend on the type of port (DUT input or output) and on the chosen kind of measurement.

They are listed in the table below.

Table 21 Default and Available Pseudo Segments

Kind of Measurement	DUT Data Input Port	DUT Data Output Port
Capture Data	Pause0 also available: Pause1	Pause also available: Acquire
Error Rate Measurement	Pause0 also available: Pause1	Pause also available: Expected 0 Expected 1
Compare and Acquire Around Error	Pause0 also available: Pause1	Pause also available: Expected 0 Expected 1 Don’t Care
Compare and Capture	Pause0 also available: Pause1	Pause also available: Expected 0 Expected 1 Don’t Care

Explanation

Pseudo segments for data generator channels:

– *Pause0*: Transmit logical zero (usually low level voltage) for the specified block length.

– *Pause1*: Transmit logical one for the specified block length.

Pseudo segments for data analyzer channels:

– *Pause*: Fall asleep for the specified block length.

– *Acquire*: Capture all DUT output data.

– *Expected 0*: Consider all non-zero data as errors.

– *Expected 1*: Consider all data that are not logical one as errors.

– *Don’t Care*: Capture, but don’t compare with expected data.

2 Choose from the menu.

How to Create and Change Loops

Loops can be specified in the columns at the right-hand side of the blocks. The right-most column is reserved for infinite loops.

A system equipped with the E4805A/B clock module provides 5 loop levels.

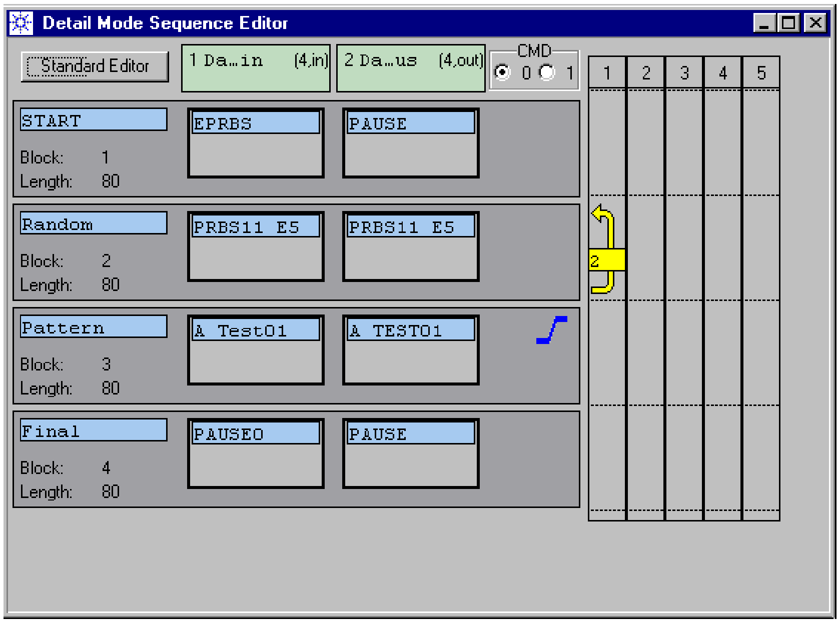
How to Create a Loop

To create a loop:

1 Click with the left mouse button into one of the columns.

Alternatively, you can also open the context menu of an empty column, choose *New Loop* and confirm the default loop properties.

This usually creates a one-block loop with 2 iterations. In the right-hand column, however, it creates an infinite loop.



Block: 1
Length: 80

Block: 2
Length: 80

Block: 3
Length: 80

Block: 4
Length: 80

Figure 192 One-Block Loop

How to Change a Loop

Loops can be changed with the mouse and from the loop context menu.

How to Change a Loop With the Mouse

1 To change the length of a loop, click the upper or lower end of the loop and drag vertically.

2 To move a loop to a different level, click the loop and drag horizontally.

3 To change the number of repetitions of a loop, double-click the loop.

This opens the Loop Properties window. Type the desired number of repetitions and confirm.

How to Change Loop Characteristics With the Keyboard

1 Open the context menu of the loop.

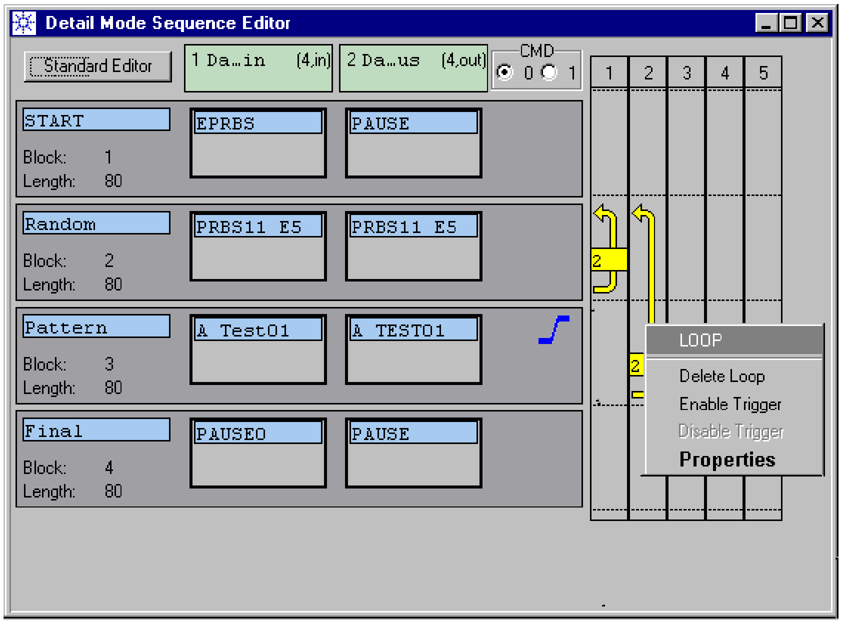


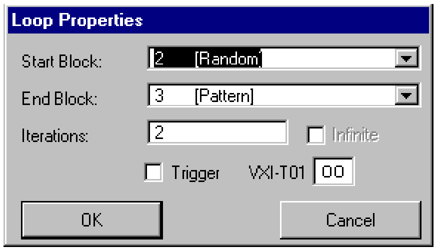
Figure 193 Loop Context Menu

2 Choose from the menu.

You can:

- Delete the loop.
- Enable/disable a trigger to be generated each time the loop is repeated.

– Select *Properties* to change the loop characteristics with the keyboard.



Loop Properties

Start Block: 2 [Random]

End Block: 3 [Pattern]

Iterations: 2 □ Infinite

□ Trigger VXI-T01 00

Figure 194 Loop Properties Window

You can change the start block, end block, number of repetitions, set a trigger, or set the VXI trigger lines (see also *"How to Change Block Properties" on page 289*).

How to Specify Events and Reactions Upon Events

The Agilent 81250 system is capable of reacting on events. The reaction can simply be a trigger pulse at the TRIGGER OUTPUT of the clock module, but also a change of the test sequence.

For general information see *"Event Handling Principles" on page 104*.

Events can be defined at any time. The reactions upon events can be specified if the Detail Mode Sequence Editor or the Data/Sequence Editor is active.

Examples can be found in *"How Do I Use Events?" on page 392*.

NOTE Event recognition is disabled when a synchronization block is executed.

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *"Controlling the Test with Events"*.

Before You Start Using Events

You can define up to 5 events for immediate action and 5 events for deferred action.

Events for immediate action are serviced as soon as they occur. Events for deferred action are serviced at the end of the block. This is in contrast to a trigger associated with a block or loop. Those actions occur when the execution of the block is started.

Events for deferred action are prioritized. The event with the highest number has the highest priority.

What You Need to Find Out Before Using Events

1. Determine what you wish to achieve. Choices are:
   - Issue a trigger pulse from the master clock module for starting/synchronizing an external instrument.
   - Set the VXI trigger lines T0/T1 for triggering other VXI modules.
   - Change the test sequence: Continue with another block, start all over, or terminate the test.

     If you wish to switch to another block, ensure that it is included in the overall sequence and labeled. You can only jump to labeled blocks.
2. Determine the release mechanism. Choices are:
   - A command issued locally by clicking a button provided by the Detail Mode and Data/Sequence Editors or remotely.
   - One or several bit combinations of the trigger pod (see also *"Trigger Pod" on page 55*).
   - The status of the VXI trigger lines TX0 and TX1 which may be changed by an external VXI module.
   - A bitstream error detected by one of the data generator/analyzer modules (not available in capture-only or BER mode).

3 Decide on the priorities.

   - Do you need immediate reaction?
   - In case of deferred reaction: Which event must be serviced under all circumstances? What is the minimum block length to guarantee reaction at the end of the block?

What You Need to Consider Before Using Events

There is of course a delay between the occurrence of an event and its recognition. There is also a delay between the recognition of an event and the reaction on that event.

Detection and Reaction Times

Detection of and reaction on events is controlled by an internal sequencer clock. The port-dependent frequency of that clock is:

$\mathtt{Clk}_{\mathtt{p}}\mathtt{\_freq = system\ clock\ frequency\ /\ segment\ resolution}$

The maximum sequencer clock frequency is hence 41.67 MHz, corresponding to a period of 24 ns. If you had set a system clock rate of 100 MHz and a segment resolution of 4, the sequencer clock frequency would be 25 MHz, corresponding to a period of 40 ns.

A system equipped with E4832A and E4861A modules has the following delays:

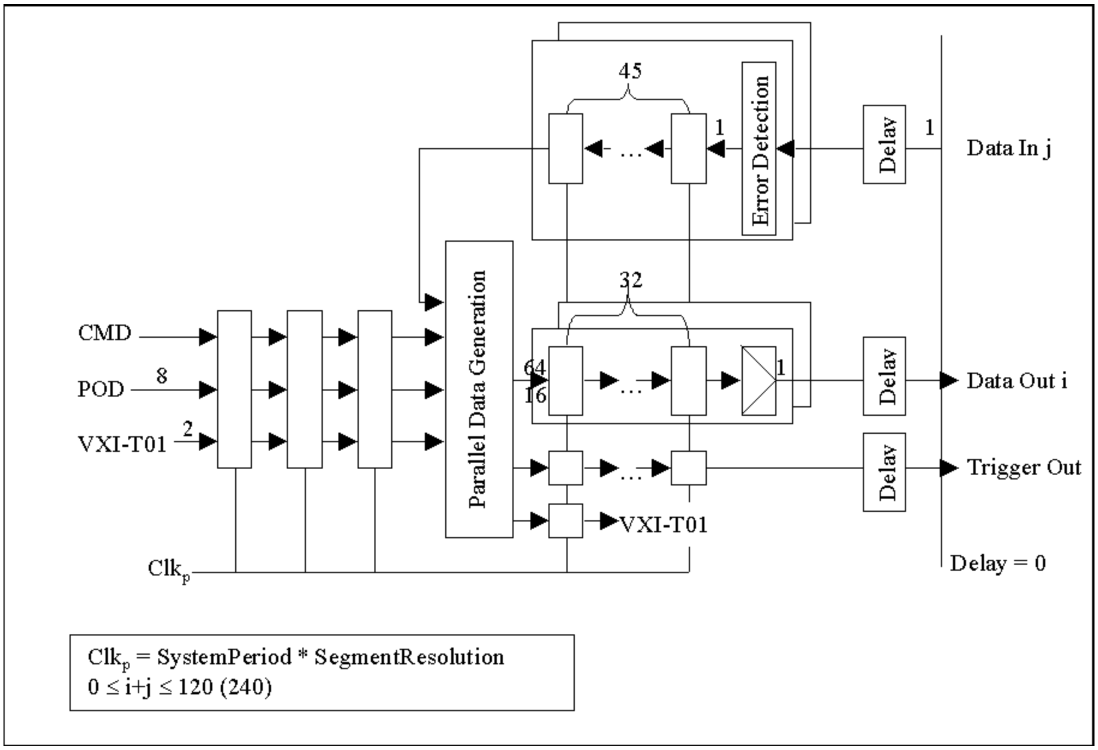


Figure 195 Delays Between Event Occurrence, Detection, and Reaction (System with E4832A, E4861A)

Explanation

Event commands as well as changes of trigger pod inputs or VXI trigger lines are processed after three periods of the sequencer clock Clk_p.

If bit stream errors (detected by one of the analyzer frontends) are included into an event definition, event processing takes 45 periods of the sequencer clock Clk_p if the system includes E4832A or E4861A modules.

Once the 11-bit event pattern has been set up, the VXI trigger signals (if specified) are set after one period of the sequencer clock.

Depending on the data module and the current segment resolution, up to 16 or 64 bits can be processed during one period of the sequencer clock.

Launching a trigger at the master clock module’s TRIGGER OUTPUT requires additional 32 periods of the sequencer clock Clk_p at a system equipped with E4832A or E4861A modules.

Switching to another sequence block needs 33 periods.

NOTE

Especially if you wish to react on errors, this behavior has to be taken into account. For triggering on errors, you should use the Compare and Capture mode, because Compare and Acquire Around Error stops automatically some time after an error and may terminate the test before you get a reaction (see also *"Choosing the Kind of Measurement" on page 261*).

Minimum Block Length

The reactions on events are associated with blocks.

If you wish to react on an error by changing the sequence or setting a trigger and the respective block is **looped**, it must have a length of at least

```
79 × segment resolution
```

to ensure that the system can react during the next repetition of the block. 79 is the sum of 46 plus 33.

If the respective block is **not looped**, it must contain more data than compared. Proper triggering or sequence changing on errors during block execution is only ensured, if the block contains

```
79 × segment resolution
```

more generated vectors than are compared. Different segments may be needed for generated and expected data.

Example

If you have a system equipped with E4832A modules, a system clock period of 10 ns (100 MHz), a segment resolution of 4, and a block length of 400, then error events within the first 4 vectors can directly lead to a trigger signal or sequence change while or after the block is executed (minimum length for sequence change is 79 × 4 = 316).

As the sequencer clock rate is 100 MHz / 4 = 25 MHz, the delay between error recognition and sequence change is 79 × 4o ns = 3.16 ms. Later error events (resulting from vectors 5 to 400) are only processed if the block is repeated, because the reaction on events is associated with the block.

To react on all possible errors within the execution of 400 vectors, the block must have a length of at least 716 vectors, and the last 316 vectors must not be evaluated.

Triggers

If you intend to issue triggers, please note: Triggers have certain characteristics. The characteristics of the master clock module's TRIGGER OUTPUT are part of the global system parameters (see *"How to Set the Characteristics of the Trigger Output" on page 192*).

The width of the trigger at the clock module or the VXI bus corresponds to the period of the internal sequencer clock:

$\mathtt{Clk_p\_period = system\ clock\ period} \times \mathtt{general\ segment\ resolution}$

Example: If you had set a system clock rate of 100 MHz and a segment resolution of 4, the trigger width would be 40 ns.

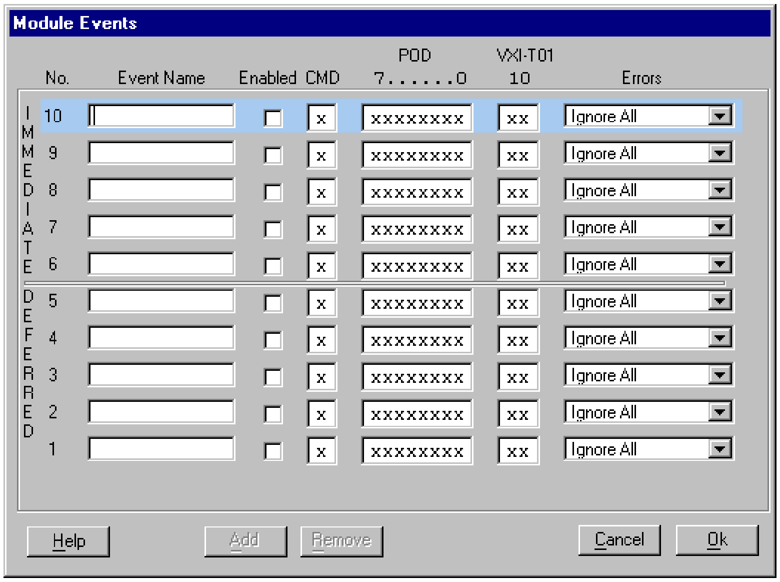
How to Define Events

To specify events:

1 Open the context menu of a block and choose *Edit Events*.

Alternatively, you can also choose *Events* from the Edit menu and select *Edit*. Actually, event definitions are independent of the chosen block.

The Module Events window appears.



	No.	Event Name	Enabled	CMD	POD 7......0	VXI-T01 10	Errors
IMMEDIATE	10		☐	x	xxxxxxxx	xx	Ignore All
	9		☐	x	xxxxxxxx	xx	Ignore All
	8		☐	x	xxxxxxxx	xx	Ignore All
	7		☐	x	xxxxxxxx	xx	Ignore All
	6		☐	x	xxxxxxxx	xx	Ignore All
DEFERRED	5		☐	x	xxxxxxxx	xx	Ignore All
	4		☐	x	xxxxxxxx	xx	Ignore All
	3		☐	x	xxxxxxxx	xx	Ignore All
	2		☐	x	xxxxxxxx	xx	Ignore All
	1		☐	x	xxxxxxxx	xx	Ignore All

Figure 196 Module Events Window

2 Choose the type (deferred/immediate) and the priority of the event.

The types are explained in *“What is an Event?” on page 105*.

3 Enter the *Event Name*.

Every event requires its own, unique name.

4 Enable the event.

Click the corresponding checkbox or move with Tab and press the space bar.

5 Select and edit the details.

These items are logically ANDed. That means, the combination of whatever is activated and detected will cause an action.

- The *CMD* column refers to the manual or remote command that can cause an interrupt.

  Manual interrupts can be produced from the Detail Mode Sequence Editor and the Data/Sequence Editor windows by clicking *CMD0* or *CMD1*.

  Remote interrupts can be produced by the test program.

  Acceptable input values are x (don't care), 0, or 1.

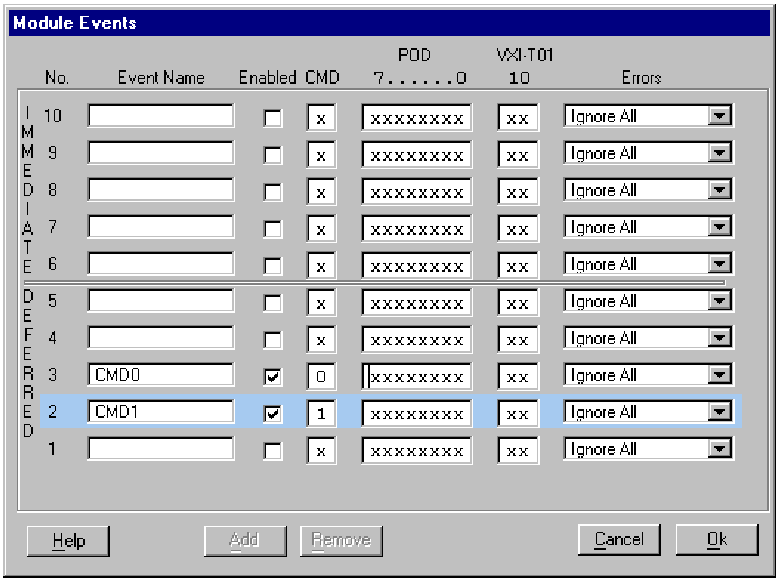
- The *POD* column refers to the trigger pod (see also *"Trigger Pod" on page 55*). You can set the expected bits to x (don't care), 0, or 1.

- The *VXI* column refers to the VXI trigger lines T0/T1. Acceptable inputs are x (don't care), 01, 11, 10. Note: If you don't wish to react on their status, ensure they are set to xx.

  You can then set the VXI trigger lines as an answer to an event.

- The *Errors* column refers to the built-in analyzer channels. Open the pulldown menu and choose from the list.

A very simple event table which activates just the command-controlled events might look as shown below.



	No.	Event Name	Enabled	CMD	POD 7......0	VXI-T01 10	Errors
IMMEDIATE	10		☐	x	xxxxxxxx	xx	Ignore All
	9		☐	x	xxxxxxxx	xx	Ignore All
	8		☐	x	xxxxxxxx	xx	Ignore All
	7		☐	x	xxxxxxxx	xx	Ignore All
	6		☐	x	xxxxxxxx	xx	Ignore All
DEFERRED	5		☐	x	xxxxxxxx	xx	Ignore All
	4		☐	x	xxxxxxxx	xx	Ignore All
	3	CMD0	☑	0	xxxxxxxx	xx	Ignore All
	2	CMD1	☑	1	xxxxxxxx	xx	Ignore All
	1		☐	x	xxxxxxxx	xx	Ignore All

Figure 197 **Simple Event Table**

The event CMD0 occurs as soon as the *CMD* radio button of the Detail Mode Sequence Editor or the Data/Sequence Editor is identified as zero or after issuing the corresponding firmware command. Similarly, the event CMD1 occurs as soon as the *CMD* radio button of the Detail Mode Sequence Editor or the Data/Sequence Editor is identified as one.

Both are deferred events, which means that the system will react as soon as the presently executed sequence block has come to its end (assuming it is either repeated or long enough, see *“What You Need to Consider Before Using Events” on page 300*).

6 When you are done, click OK.

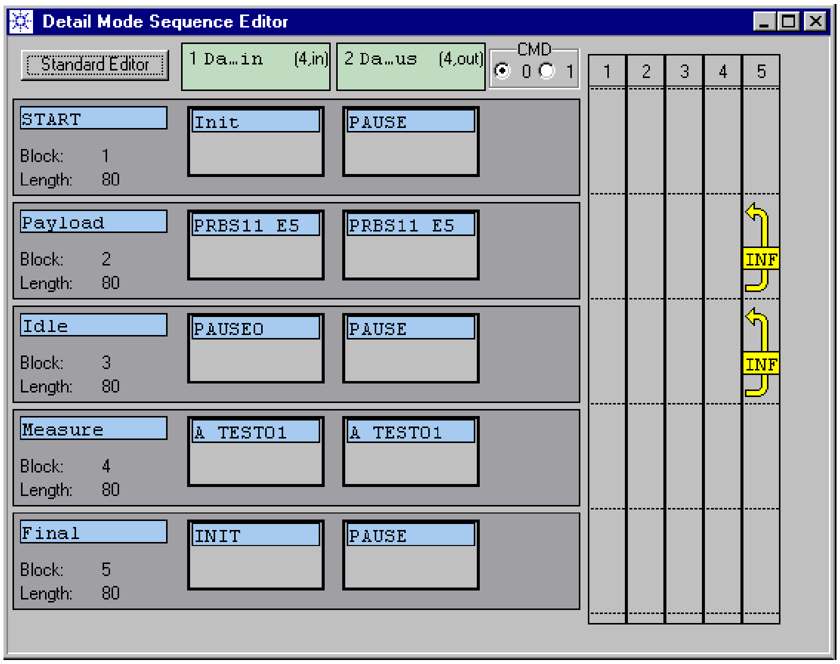
How to Specify the Reactions on Events

The reactions on events are block-related. You can specify individual reactions for each block of the sequence.

An example may be helpful to understand this procedure. It builds up on the event definition example shown above.

Stop and Go Example

We have set up the sequence shown below:



		1 Da…in (4,in)	2 Da…us (4,out)
START	Block: 1 Length: 80	Init	PAUSE
Payload	Block: 2 Length: 80	PRBS11 E5	PRBS11 E5
Idle	Block: 3 Length: 80	PAUSE0	PAUSE
Measure	Block: 4 Length: 80	A TEST01	A TEST01
Final	Block: 5 Length: 80	INIT	PAUSE

Figure 198 **Sequence for the Stop and Go Example**

We wish to run the Payload block until the result stabilizes. Then, upon a command, the test shall pause, so that we can examine the results.

A second command shall cause the test to continue with the Measure block and to finish.

We have defined the events CMD0 and CMD1 as shown in the example of *"How to Define Events" on page 303*.

How to Fill In the Branch Table

Each block has its own branch table.

1 Open the context menu of block 2.

2 Choose *View / Edit Branches*.

If no reactions have been specified so far, an empty branch table appears.

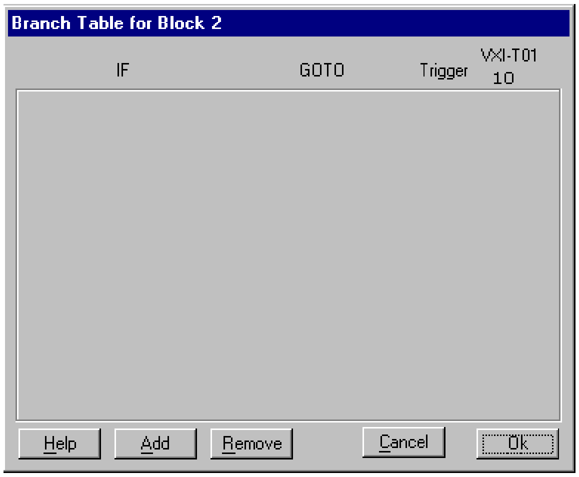


Figure 199 Empty Branch Table

3 Click the *Add* button.

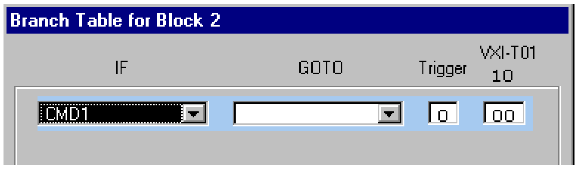


Figure 200 Branch Table Structure

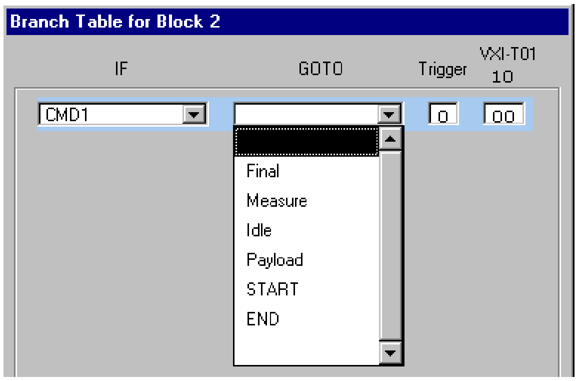
Now you can see how the table is built up: If the specified event has occurred,

- go to a certain block of the sequence and/or
- output a trigger at the clock module and/or
- set the VXI trigger lines.

4 In the *IF* column, select one of the available events. The display starts with the lowest-priority event that has been defined (see also *"How to Define Events" on page 303*).

The pull-down menu offers also the DEFAULT event. This is a deferred event which occurs at the end of a block and must not be defined. It has a lower priority than any user-defined event and can be used to change the normal flow (which is either continuation or iteration).

5 In the *GOTO* column, select one of the available blocks.



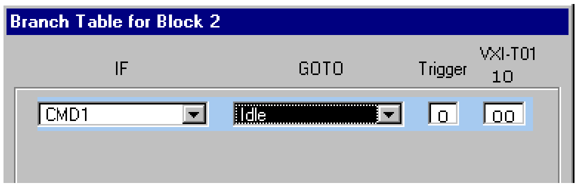
Branch Table for Block 2

IF	GOTO	Trigger	VXI-T01 10
CMD1		0	00

Figure 201 Branch Table – Block Selection

Note that the END block is an implicit block which is always available. It terminates the sequence.

The branch table of block 2 in our example is shown below:



Branch Table for Block 2

IF	GOTO	Trigger	VXI-T01 10
CMD1	Idle	0	00

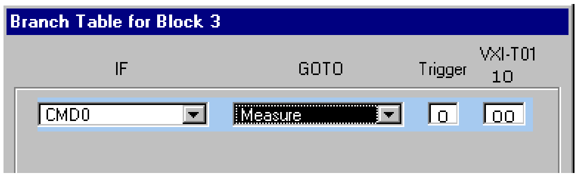
Figure 202 Branch Table – Block 2

As soon as CMD1 occurs, the block will not be repeated any more. The sequence will execute the Idle block.

6 Decide whether you wish to generate a trigger pulse. If you wish to activate the VXI triggers for controlling other VXI modules, ensure that you have not defined an event based on the status of the VXI trigger lines.

7 Click *OK*.

To complete the example, you have to repeat steps 1 to 6 for block 3.

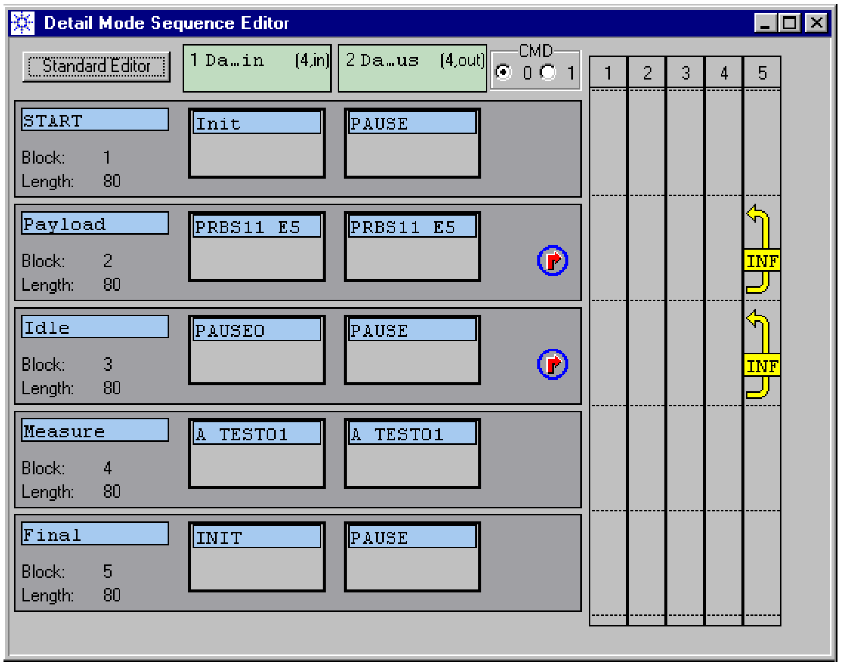

Branch Table for Block 3

IF	GOTO	Trigger	VXI-T01 10
CMD0	Measure	0	00

Figure 203 Branch Table – Block 3

As soon as CMD0 occurs, the infinite loop of the Idle block will be left, and the sequence will execute the Measure block

The sequence now shows that branches have been inserted.



Detail Mode Sequence Editor

START	Block: 1	Length: 80	Init	PAUSE
Payload	Block: 2	Length: 80	PRBS11 E5	PRBS11 E5
Idle	Block: 3	Length: 80	PAUSE0	PAUSE
Measure	Block: 4	Length: 80	A TEST01	A TEST01
Final	Block: 5	Length: 80	INIT	PAUSE

Figure 204 Sequence With Branches on Events

If you now run the test, you can terminate the Payload test by clicking the CMD 1 radio button. You can then resume the test by clicking the CMD 0 radio button. Block 4 and block 5 will be executed, and the test will finish.

More examples can be found in *“How Do I Use Events?” on page 392*.

Creating and Editing Segments

The stream of generated and expected data is defined by the data sequence. A sequence is built up of blocks. Each block references one data segment for each DUT data port.

The two different types of data segments are pseudo segments and real segments (see *“Data Segments” on page 81*).

Real segments can be created and modified manually. This chapter explains how this is done. See:

- *“How to Create a New Segment” on page 310*
- *“How to Edit a Stored Segment” on page 321*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial*.

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *“Creating the Test Data Pattern”*.

How to Create a New Segment

Two types of real segments can be created and edited with the Segment Editor:

- PRBS/PRWS segments

  PRBS/PRWS segments contain pseudo random data in bit stream (PRBS) or word stream (PRWS) format. Pseudo random data is defined by the generating polynomial.

- Memory segments.

  Memory segments contain a user-defined data pattern.

NOTE Any new or modified segment needs to be saved on disk before it can be referenced in a block.

For details see:

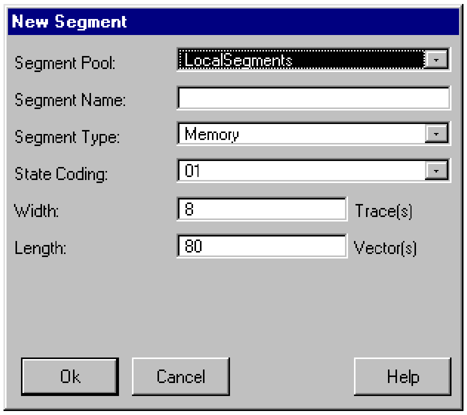
- *"How to Start Creating a New Segment" on page 311*
- *"How to Create a Memory Segment" on page 312*
- *"How to Create a PRBS/PRWS Segment" on page 319*
- *"How to Save a New or Changed Segment" on page 321*

How to Start Creating a New Segment

To start the New Segment dialog:

1 Open the *File* menu and choose *New Segment*.

You start with the defaults as shown in the figure below.



Segment Pool: LocalSegments

Segment Name:

Segment Type: Memory

State Coding: 01

Width: 8 Trace(s)

Length: 80 Vector(s)

Figure 205 New Segment Window

If the Detail Mode Sequence Editor or the Data/Sequence Editor is displayed, you can also open the context menu of the segment you wish to replace and choose *New Segment*. If you enter the New Segment dialog this way, the block and port characteristics are already filled in.

2 Choose the *Segment Pool*.

Segments in the global segment pool can be accessed from all settings. Segments in the local segment pool can only be accessed from the current setting.

3 Enter or edit the *Segment Name*.

TIP Enter a name that explains the contents or purpose of the segment.

4 Choose the *Segment Type*.

Type “Memory” means that a freely programmable pattern is stored in the database.

Type “PRBS” or “PRWS” means that an algorithm is used for generating a pseudo random bit or word stream. You can choose between pure and distorted PRxS. Pure PRxS data is generated at runtime by hardware feedback shift registers built into the modules.

Type “SFI5” is a special type of a pure PRBS/PRWS segment. It specifies pseudo random data that is formatted according to the SFI-5 standard. This type of segment can only be used for E4861B modules. These modules are capable of generating formatted random data. For details see the manual *Testing SFI-5 Devices*.

NOTE PRWS segments are mainly used for testing multiplexers/demultiplexers. The first terminal of the DUT input port (counted from top to bottom as shown in the Connection Editor) gets the first state of the generated random sequence, followed by the next lower pin, and so on. If you had set up a 4-bit port and bit count starts with one, the first terminal would receive bit 1, 5, 9, and so on.

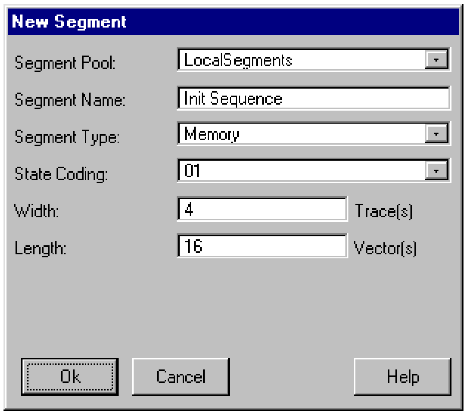
This fashion of sending the data to the channels allows you to connect the port terminals to the frontends in arbitrary order. You need only take care that your physical cable connections correspond exactly to the connections shown in the Connection Editor.

For details see also *"Appendix B: PRBS/PRWS Data Segments" on page 425*.

How to Create a Memory Segment

Start the New Segment dialog (see *"How to Start Creating a New Segment" on page 311*).

Once you have decided to create a new memory segment and where to store it, fill in the remaining fields:



New Segment

Segment Pool: LocalSegments

Segment Name: Init Sequence

Segment Type: Memory

State Coding: 01

Width: 4 Trace(s)

Length: 16 Vector(s)

Figure 206 New Segment Window for Creating a Memory Segment

1 Specify the *State Coding*. Choices are 01 or 0x1.

The state coding 01 specifies that every bit of the segment occupies one bit in memory. This is adequate for all data segments to be downloaded to generator channels.

The state coding 0x1 can be used for expected data. This coding enables the x-character used to denote don't care bits. State coding

0x1 specifies that each bit of the segment occupies two bits in memory.

2 Set the *Width* and *Length* of the new data segment.

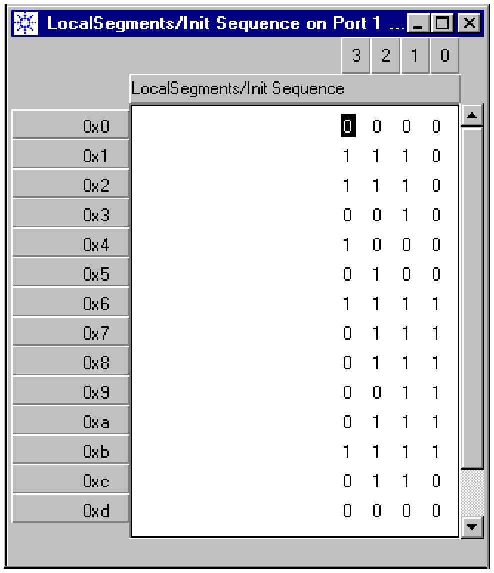
The width represents the number of pins included in a port and defines the number of traces.

The length is the pattern length and, thus, the number of vectors.

NOTE The length may exceed but must not be shorter than the length of the block where the segment is going to be inserted.

3 Click *OK*.

This opens the Segment Editor for the newly specified memory segment.



LocalSegments/Init Sequence

	3	2	1	0
0x0	0	0	0	0
0x1	1	1	1	0
0x2	1	1	1	0
0x3	0	0	1	0
0x4	1	0	0	0
0x5	0	1	0	0
0x6	1	1	1	1
0x7	0	1	1	1
0x8	0	1	1	1
0x9	0	0	1	1
0xa	0	1	1	1
0xb	1	1	1	1
0xc	0	1	1	0
0xd	0	0	0	0

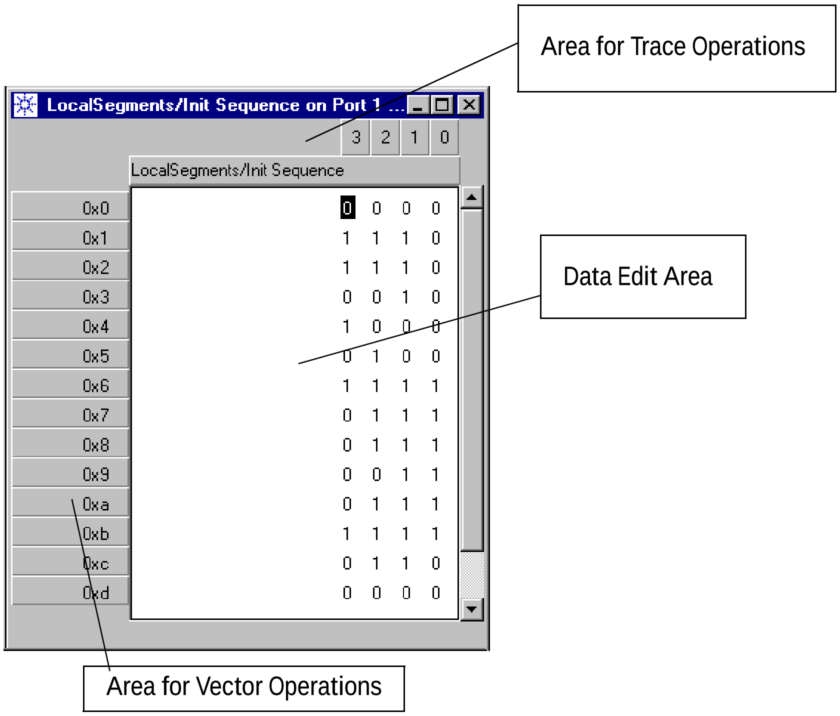
Figure 207 Segment Editor Window

The window shows the specified vectors (horizontal lines) and traces (vertical rows).

Characteristics of the Segment Editor Window

The Segment Editor window has three active areas:

- An area for vector operations
- An area for trace operations
- The data edit area



LocalSegments/Init Sequence on Port 1 ...

LocalSegments/Init Sequence

	3	2	1	0
0x0	0	0	0	0
0x1	1	1	1	0
0x2	1	1	1	0
0x3	0	0	1	0
0x4	1	0	0	0
0x5	0	1	0	0
0x6	1	1	1	1
0x7	0	1	1	1
0x8	0	1	1	1
0x9	0	0	1	1
0xa	0	1	1	1
0xb	1	1	1	1
0xc	0	1	1	0
0xd	0	0	0	0

Figure 208 Segment Editor Window Areas

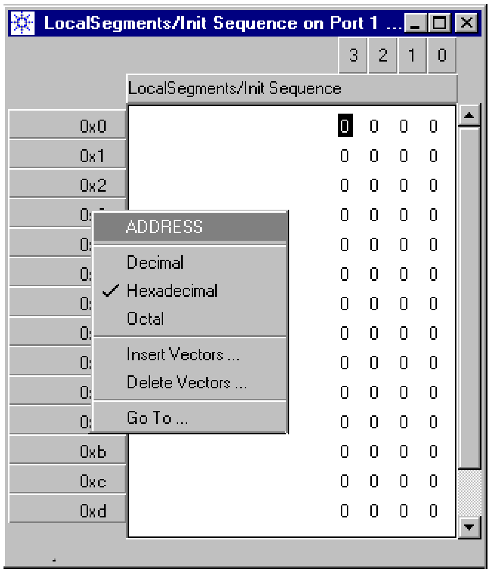
Each of these areas has its own context menu, indicated by the cursor changing its shape when placed over the areas.

How to Use the Segment Editor’s Vector Operations Area

Clicking a vector address highlights that vector.

Dragging the cursor across several vector addresses highlights a block of vectors.

The context menu is opened by clicking with the right mouse button. It provides the following options:



	3	2	1	0
0x0	0	0	0	0
0x1	0	0	0	0
0x2	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0xb	0	0	0	0
0xc	0	0	0	0
0xd	0	0	0	0

Figure 209 Segment Editor – Context Menu for Vector Operations

The available options are:

- Change the address display format (decimal, hex, or octal)
- Insert or delete highlighted vectors in the table
- Jump to a certain vector address

How to Use the Segment Editor’s Trace Operations Area

Clicking a trace number highlights that trace.

Dragging the cursor across several trace numbers highlights a block of traces.

The context menu is opened by clicking with the right mouse button. It provides the following options:

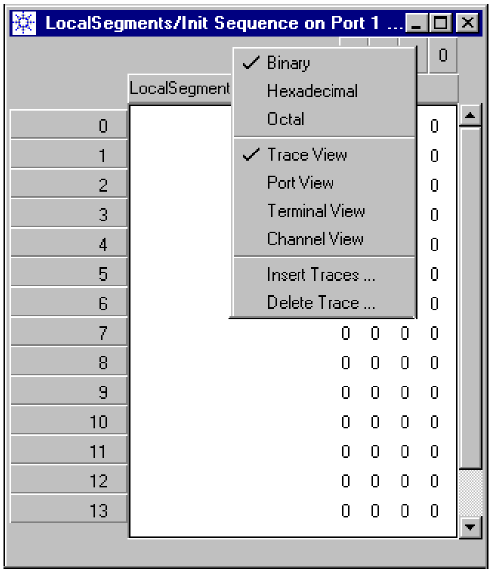


Figure 210 **Segment Editor – Context Menu for Trace Operations**

The available options are:

- Change the data display format.

  In binary mode, each trace has its own column. In hexadecimal mode, four traces are combined in one column (range 0 to F_{hex}). In octal mode, three traces are combined in one column (range 0 to 7_{oct}).

- Display the port, defined terminals or connected channels instead of trace numbers (available if you have started the Segment Editor by clicking a segment in the Sequence Editor).

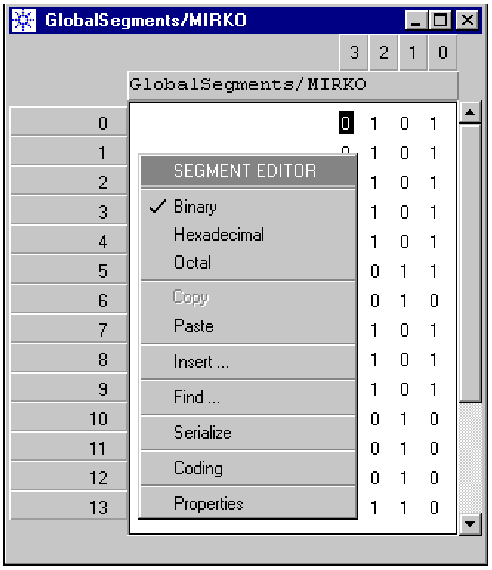
- Insert new or delete highlighted traces in the table.

How to Use the Segment Editor’s Data Edit Area

Clicking a bit highlights that bit. With the spacebar, you can toggle from zero to one and vice versa.

Dragging the cursor across several bits highlights a block.

The context menu provides the following options:



	3	2	1	0
0	0	1	0	1
1	0	1	0	1
2		1	0	1
3		1	0	1
4		1	0	1
5		0	1	1
6		0	1	0
7		1	0	1
8		1	0	1
9		1	0	1
10		0	1	0
11		0	1	0
12		0	1	0
13		1	1	0

Figure 211 Segment Editor – Initial Context Menu for Editing Data

The available options are:

- Change the data display format (decimal, octal, or hex)
- Insert or delete highlighted vectors or traces
- Find a certain pattern within the segment
- Convert a parallel segment to serial or a serial segment to parallel
- Change the state coding
- Review the segment’s setup information, but don’t change it

If you have highlighted a block of data, the context menu provides additional options.

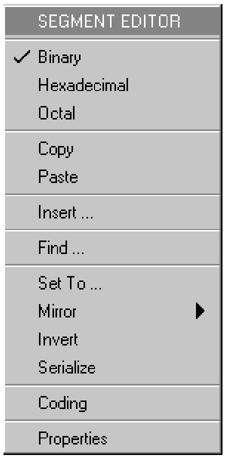


Figure 212 Segment Editor – Extended Context Menu for Editing Data

The additional options are:

- Copy the block to the clipboard (it can then be pasted somewhere else).
- Set the whole block to 1, or 0, or don’t care (the latter only if the state coding is 0x1).
- Mirror the block contents horizontally or vertically.
- Invert the bits contained in the block.

NOTE You can also paste captured data from the Error State Display. For details see *“How to Transfer Captured Data Into a Segment” on page 352*.

Segment Editor Keyboard Shortcuts and Defaults

Cursor keys The cursor movement is from left to right and from top to down. If a block is highlighted, the cursor only moves within that block.

Page Up / Down keys The Page Up and Page Down keys allow to scroll vertically through the data segment. Step size is the number of lines actually visible in the editor window.

Home key The Home key moves the cursor to the vector number 0x0 in hexadecimal (equals decimal 0) and the highest trace number.

End key The End key moves the cursor to the highest vector number and trace number 0.

Insert key The Insert key can be used to insert vectors (rows) or traces (columns) in the table.

Delete key The Delete key can be used to delete highlighted vectors or traces.

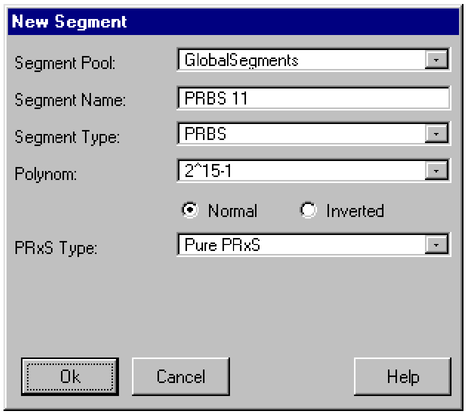
Scroll Bar The Scroll Bar at the right side of the editor window helps to position the cursor in the middle portion of a large data segment.

Esc key To deselect a selection, press the Esc key or click outside the highlighted block.

To highlight the whole segment, click the segment label.

How to Create a PRBS / PRWS Segment

Start the New Segment dialog (see *“How to Start Creating a New Segment” on page 311*). Once you have decided to create a PRBS or PRWS segment, the window changes:



Segment Pool: GlobalSegments

Segment Name: PRBS 11

Segment Type: PRBS

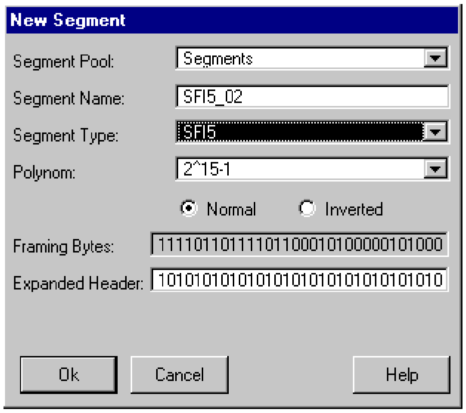
Polynom: 2^15-1

● Normal ○ Inverted

PRxS Type: Pure PRxS

Figure 213 **PRBS / PRWS Segment Specification Window**

If you have chosen the *Segment Type* SFI5, you have the following options:



New Segment

Segment Pool: Segments

Segment Name: SFI5_02

Segment Type: SFI5

Polynom: 2^15-1

● Normal ○ Inverted

Framing Bytes: 11110110111101100010100000101000

Expanded Header: 10101010101010101010101010101010

Figure 214 **SFI5 Segment Specification Window**

1 Choose one of the available *Polynomials*.

The polynomial defines the complexity of the pseudo random data. The available PRBS/PRWS polynomials are 2^5–1 through 2^{15}–1, 2^{23}–1, and 2^{31}–1.

2 Select *Normal* or *Inverted*.

If inverted is selected, the PRBS/PRWS is output in inverse mode.

3 Choose the *PRxS Type* (not applicable to SFI5 segments).

The options are:

- Pure PRxS
- Errored PRxS
- Variable Marker Density
- Extended Zeros/Ones

Pure PRxS do not consume data memory of the channels. Such data is generated at runtime by the hardware.

If you select a distorted PRBS/PRWS, the polynomial defines also its memory consumption. A distorted 2^{15}–1 PRBS/PRWS, for example, uses 32767 words of memory.

For the PRBS/PRWS polynomials 2^{23}–1 and 2^{31}–1 only pure PRxS is supported.

If you have chosen a non-pure PRxS, an additional parameter needs to be set. For details see *“Appendix B: PRBS/PRWS Data Segments” on page 425*.

4 *Framing Bytes* and *Expanded Header* are only applicable to SFI5 segments. They refer to the SFI-5 data transmission protocol. You can edit the *Expanded Header*, if desired. For details see *Some Characteristics of SFI-5* in the manual *Testing SFI-5 Devices*.

5 Click *OK* to finish creating the PRBS/PRWS segment.

How to Save a New or Changed Segment

To save a new or changed segment:

1 Open the *File* menu

2 Select *Save Segment* to save the segment under its original name.

Alternatively, you can also click *Save Segment As* to save the segment under a new name.

See also *"Save Segment" on page 155* and *"Save Segment As" on page 155*.

How to Edit a Stored Segment

To view or edit a stored segment, you have to select that segment from the pool of segments.

For details see:

"How to Select a Segment" on page 322

"How to Edit a Memory Segment" on page 323

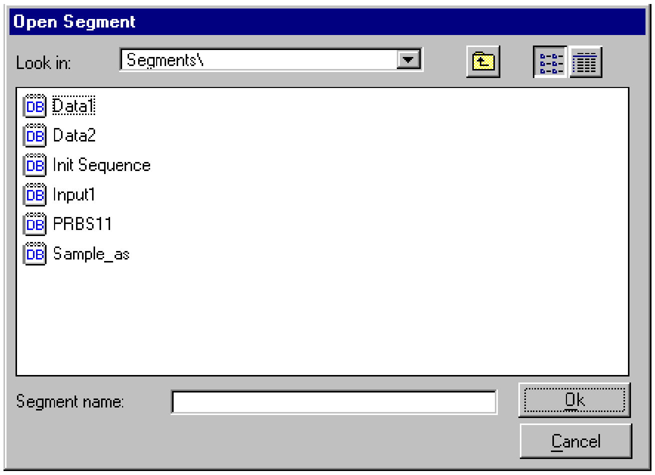
"How to Edit a PRBS/PRWS Segment" on page 333

How to Select a Segment

1 Click the Segment Editor icon.

Alternatively, you can also choose *Open Segment* from the *File* menu.

The Open Segment window appears:



Look in: Segments\

Data1
Data2
Init Sequence
Input1
PRBS11
Sample_as

Figure 215 Open Segment Window

2 Select the Segment you wish to view or edit.

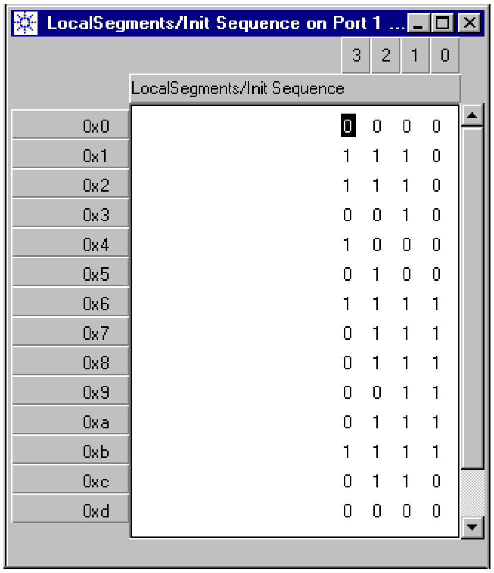
3 Click OK.

This opens the Segment Editor.

NOTE You can also open the Segment Editor directly by double-clicking a segment in the Detail Mode Sequence Editor or Data/Sequence Editor.

How to Edit a Memory Segment

After selecting a memory segment, the Segment Editor shows the data pattern:



LocalSegments/Init Sequence on Port 1 ...

LocalSegments/Init Sequence

	3	2	1	0
0x0	0	0	0	0
0x1	1	1	1	0
0x2	1	1	1	0
0x3	0	0	1	0
0x4	1	0	0	0
0x5	0	1	0	0
0x6	1	1	1	1
0x7	0	1	1	1
0x8	0	1	1	1
0x9	0	0	1	1
0xa	0	1	1	1
0xb	1	1	1	1
0xc	0	1	1	0
0xd	0	0	0	0

Figure 216 Segment Editor Window

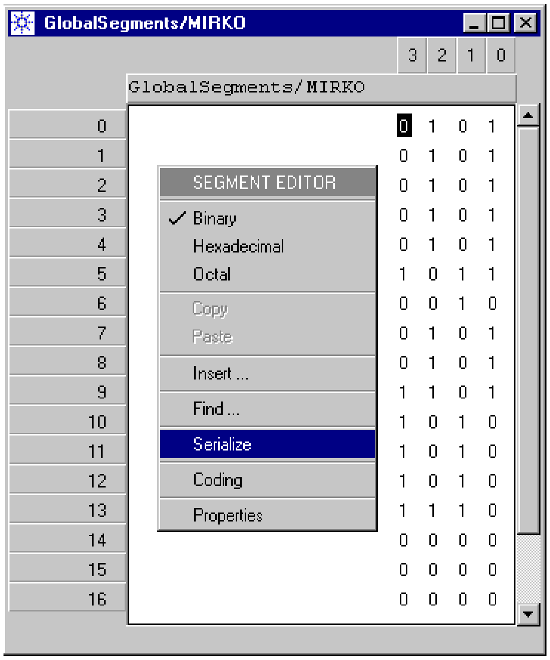
For details on how to operate the Segment Editor see *“Characteristics of the Segment Editor Window” on page 314.*

How to Convert a Memory Segment from Parallel to Serial

This function largely supports the setup of multiplexer tests.

If you have opened a parallel segment (a segment holding more than one trace), you can convert its contents to serial format (a segment holding just one trace).

1 Open the context menu of the data edit area and choose *Serialize*.

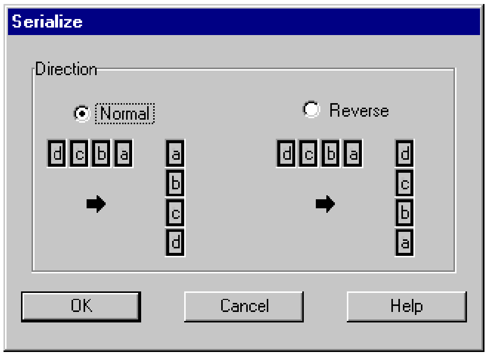


	3	2	1	0
0	0	1	0	1
1	0	1	0	1
2	0	1	0	1
3	0	1	0	1
4	0	1	0	1
5	1	0	1	1
6	0	0	1	0
7	0	1	0	1
8	0	1	0	1
9	1	1	0	1
10	1	0	1	0
11	1	0	1	0
12	1	0	1	0
13	1	1	1	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0

Figure 217 Context Window for Editing a Parallel Segment

Alternatively, you can also choose *Serialize* from the *Tools* menu.

2 Select the way you want to sort the data (*Normal* or *Reverse*).



Serialize

Direction

● Normal ○ Reverse

Figure 218 Serialize Window

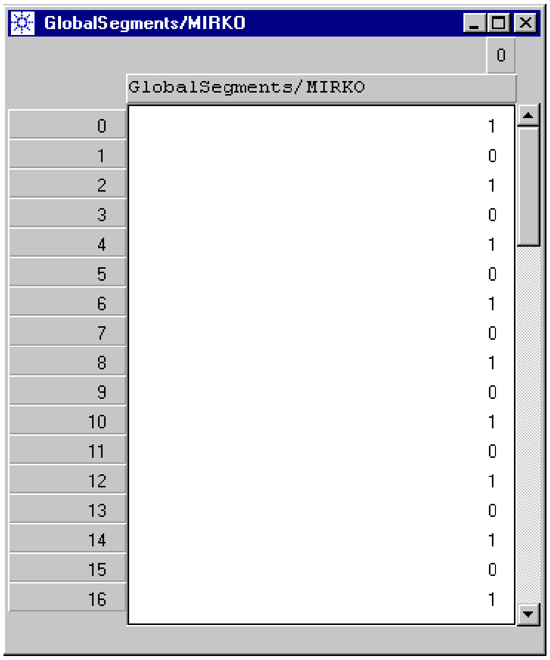
Each vector forms a data word. Trace 0 usually holds the LSB. The leftmost trace holds the MSB.

– Normal: The data is sorted from LSB to MSB.

– Reverse: The data is sorted from MSB to LSB.

3 Click *OK*.

The following figure shows the result of a “normal” conversion:



GlobalSegments/MIRKO

	0
	GlobalSegments/MIRKO
0	1
1	0
2	1
3	0
4	1
5	0
6	1
7	0
8	1
9	0
10	1
11	0
12	1
13	0
14	1
15	0
16	1

Figure 219 Serialized Segment

The segment length is now the original length × the number of traces.

4 Save the segment (see *“How to Save a New or Changed Segment” on page 321*).

How to Convert a Memory Segment from Serial to Parallel

This function largely supports the setup of demultiplexer tests.

If you have opened a serial segment (a segment holding just one trace), you can convert its contents to parallel format (a segment holding several traces).

1 Open the context menu of the data edit area and choose *Deserialize*.

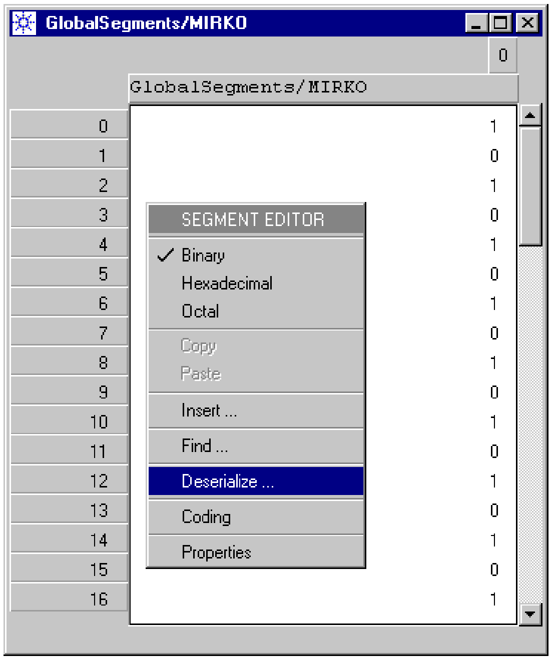
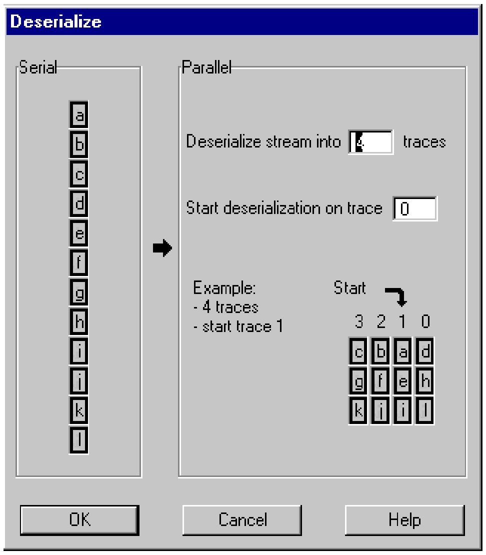


Figure 220 Context Window for Editing a Serial Segment

Alternatively, you can also choose *Deserialize* from the *Tools* menu.

2 Enter the number of traces you wish to split the data into.



Deserialize

Serial

Parallel

Deserialize stream into 4 traces

Start deserialization on trace 0

Example:
- 4 traces
- start trace 1

Figure 221 Deserialize Window

The number of traces has to be an integer fraction of the segment length. All the resulting traces will be of equal length. The converter does not patch any missing data.

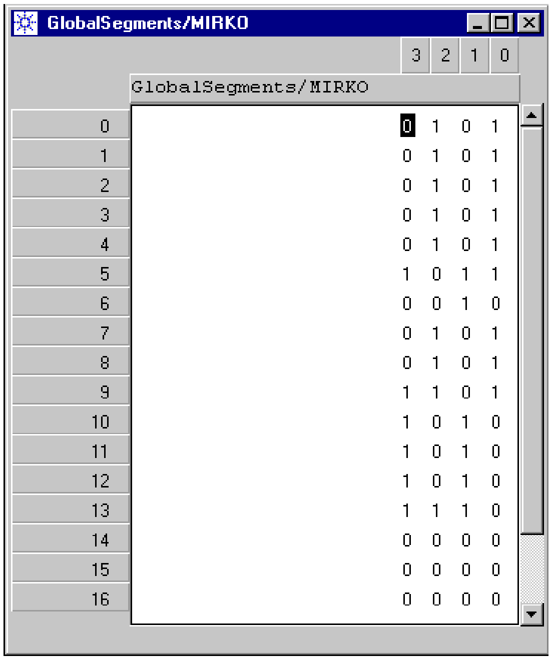
3 Set the start position of the first bit.

You can start the deserialization at any of the specified traces. The example shown in the window illustrates what happens, if you do not start at trace 0.

Setting an individual start position allows you to adapt the conversion to a demultiplexer that does not provide the first bit of the serial data stream at the pin connected to trace 0.

4 Click *OK*.

The following figure shows the result of a conversion into four traces, starting at trace 0:



GlobalSegments/MIRKO

GlobalSegments/MIRKO

	3	2	1	0
0	0	1	0	1
1	0	1	0	1
2	0	1	0	1
3	0	1	0	1
4	0	1	0	1
5	1	0	1	1
6	0	0	1	0
7	0	1	0	1
8	0	1	0	1
9	1	1	0	1
10	1	0	1	0
11	1	0	1	0
12	1	0	1	0
13	1	1	1	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0

Figure 222 Deserialized Segment

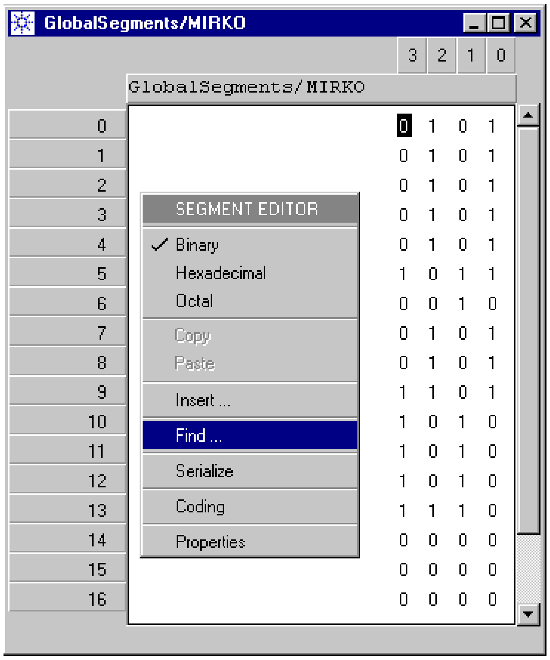
The segment length is now its original length divided by the number of traces.

5 Save the segment (see *"How to Save a New or Changed Segment" on page 321*).

How to Locate a Data Pattern Within a Segment

With the Find function, you can locate data patterns contained in a memory segment.

1 Open the context menu of the data edit area and choose *Find*.



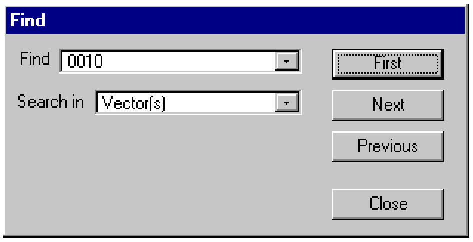
GlobalSegments/MIRKO

	3	2	1	0
0	0	1	0	1
1	0	1	0	1
2	0	1	0	1
3	0	1	0	1
4	0	1	0	1
5	1	0	1	1
6	0	0	1	0
7	0	1	0	1
8	0	1	0	1
9	1	1	0	1
10	1	0	1	0
11	1	0	1	0
12	1	0	1	0
13	1	1	1	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0

Figure 223 Context Window for Editing a Segment

Alternatively, you can also choose *Find* from the *Edit* menu.

2 Enter the pattern you wish to locate.



Find: 0010

Search in: Vector(s)

Figure 224 Find Window

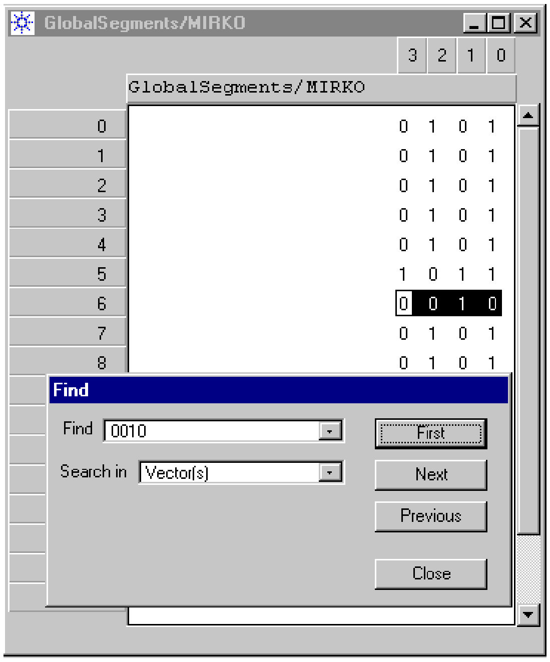
Depending on the current data representation, you can search for binary, hexadecimal, or octal patterns.

3 Decide on searching in vectors or traces.

If you are searching vectors, the length of the search pattern must not exceed the width of the segment.

If you are searching traces, the length of the search pattern must not exceed the length of the segment.

4 Click *First*.



GlobalSegments/MIRKO

	3	2	1	0
0	0	1	0	1
1	0	1	0	1
2	0	1	0	1
3	0	1	0	1
4	0	1	0	1
5	1	0	1	1
6	0	0	1	0
7	0	1	0	1
8	0	1	0	1

Find

Find: 0010

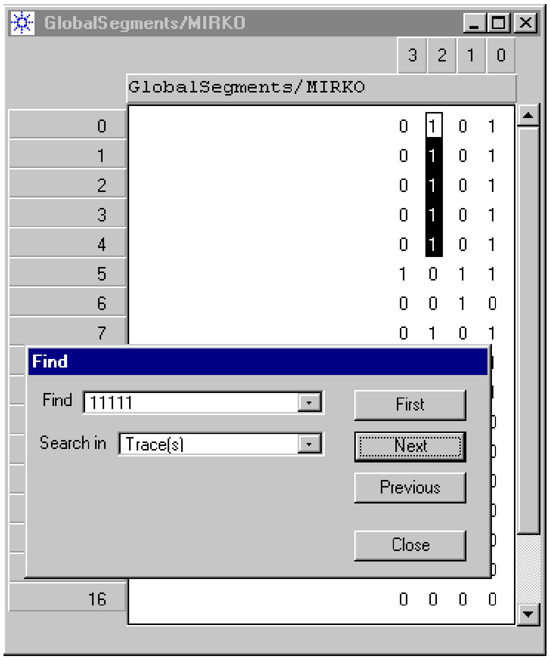
Search in: Vector(s)

Figure 225 Searching Vectors

The first occurrence of the pattern is highlighted.

5 Click *Next* to search for the next occurrence, or *Previous* to move backward.

Searching traces returns the following result:



GlobalSegments/MIRKO

	3	2	1	0
0	0	1	0	1
1	0	1	0	1
2	0	1	0	1
3	0	1	0	1
4	0	1	0	1
5	1	0	1	1
6	0	0	1	0
7	0	1	0	1
16	0	0	0	0

Find

Find: 11111

Search in: Trace(s)

Figure 226 Searching Traces

Search direction

The search direction depends on whether you search through vectors or traces (see also the figure below):

- Search vectors: Starts at vector 0, leftmost trace. Continues from left to right. After trace 0 is reached, the search proceeds to the next vector, leftmost trace.
- Search traces: Starts at leftmost trace, vector 0. Continues from top to bottom. After the last vector is reached, the search proceeds to the next lower trace.

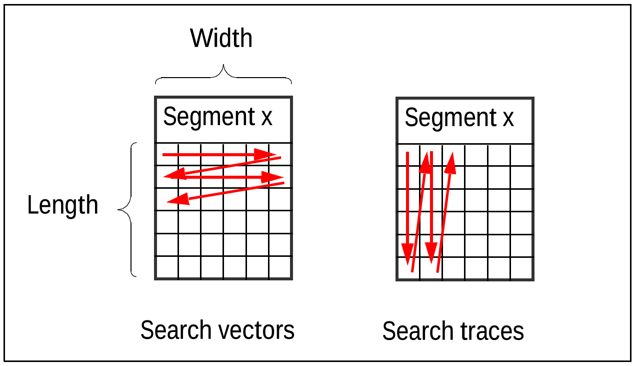
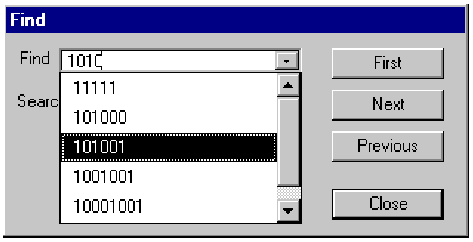


Figure 227 Search Algorithm

Search pattern memory

When you open the Find dialog, you may wish to search for a pattern you have already searched for in a previous session.

To support this, the software keeps a list of the previously used patterns:


Find

Find 101(

Searc

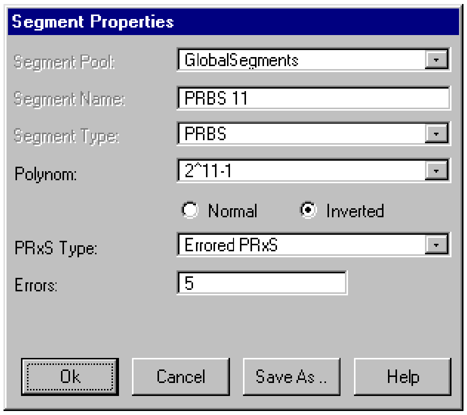
11111
101000
101001
1001001
10001001

Figure 228 List of Previously Used Patterns

Simply choose from that list.

How to Edit a PRBS / PRWS Segment

After selecting a PRBS/PRWS segment, the Segment Properties window shows the specification:



Segment Properties

Segment Pool: GlobalSegments

Segment Name: PRBS 11

Segment Type: PRBS

Polynom: 2^11-1

○ Normal ● Inverted

PRxS Type: Errored PRxS

Errors: 5

Figure 229 PRBS / PRWS Segment Properties Window

You can now change the parameters. For details see *"How to Create a PRBS/PRWS Segment" on page 319*.

Using the Data / Sequence Editor

The Data/Sequence Editor combines the functions of the Detail Mode Sequence Editor and the Segment Editor.

The Data/Sequence Editor can be used to:

- Inspect all details of the test sequence including data patterns
- Change the test sequence
- Change block characteristics
- Inspect and change the segments contained in the blocks

The Data/Sequence Editor is particularly useful on a system equipped with a high resolution video screen because it eliminates the need for switching between the two standard editors.

The procedures for editing data are basically the same as incorporated in the Sequence Editor and the Segment Editor.

This chapter comprises the sections:

- *"How to Start the Data/Sequence Editor" on page 336*
- *"How to Customize the Data/Sequence Display" on page 338*
- *"How to Change the Sequence or Edit Segments" on page 341*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo

♦ In the *Tutorial*, select *"How to Use the Data/Sequence Editor"*.

How to Start the Data / Sequence Editor

To start the Data/Sequence Editor:

1 Select *Data/Sequence Editor* from the *Go* menu.

Contents of the Data / Sequence Editor Window

The Data/Sequence Editor identifies:

- The DUT i/o ports that have been set up
- The terminals that have been set up within the ports
- The blocks that form the test sequence
- The block markers that indicate generated triggers, analyzer synchronization, and sequence changes on events
- The loops that repeat single or groups of blocks
- The segments that have been included in the blocks
- The data patterns stored in the segments

All this is shown in one window.

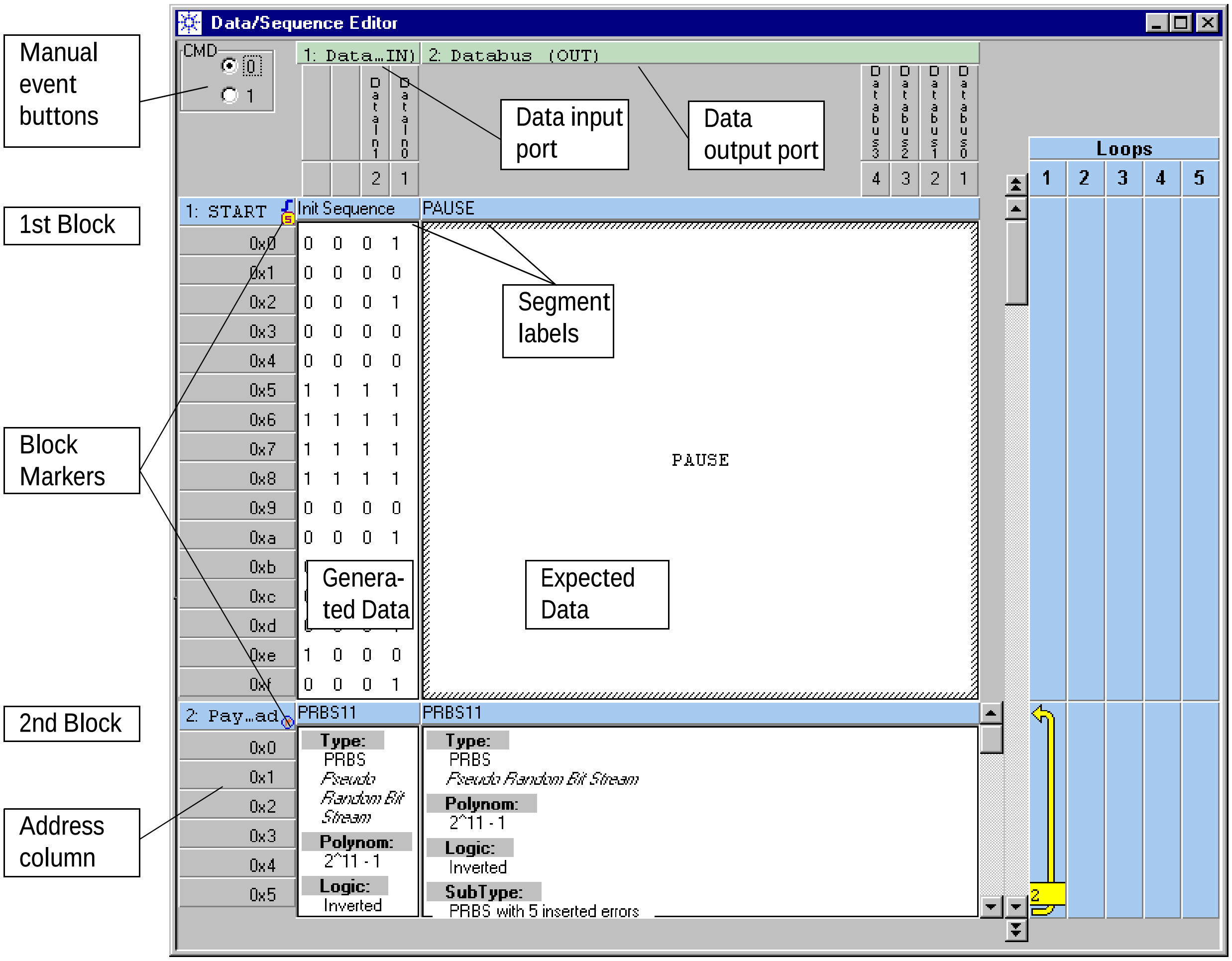


Figure 230 Data / Sequence Editor Window

NOTE Note that the Data/Sequence Editor provides vertical scroll bars for each block and an overall scroll bar for the sequence.

How to Customize the Data / Sequence Display

You can adapt the display to your preferences by changing:

- The width of the columns
- The height of the blocks
- The address display format
- The labeling of the displayed traces

How to Change the Width of the Columns

To view more or less columns (ports):

1. Move the cursor onto the vertical line that marks the column border.

   The cursor changes its shape.

2. Hold the mouse button depressed and drag the border line horizontally.

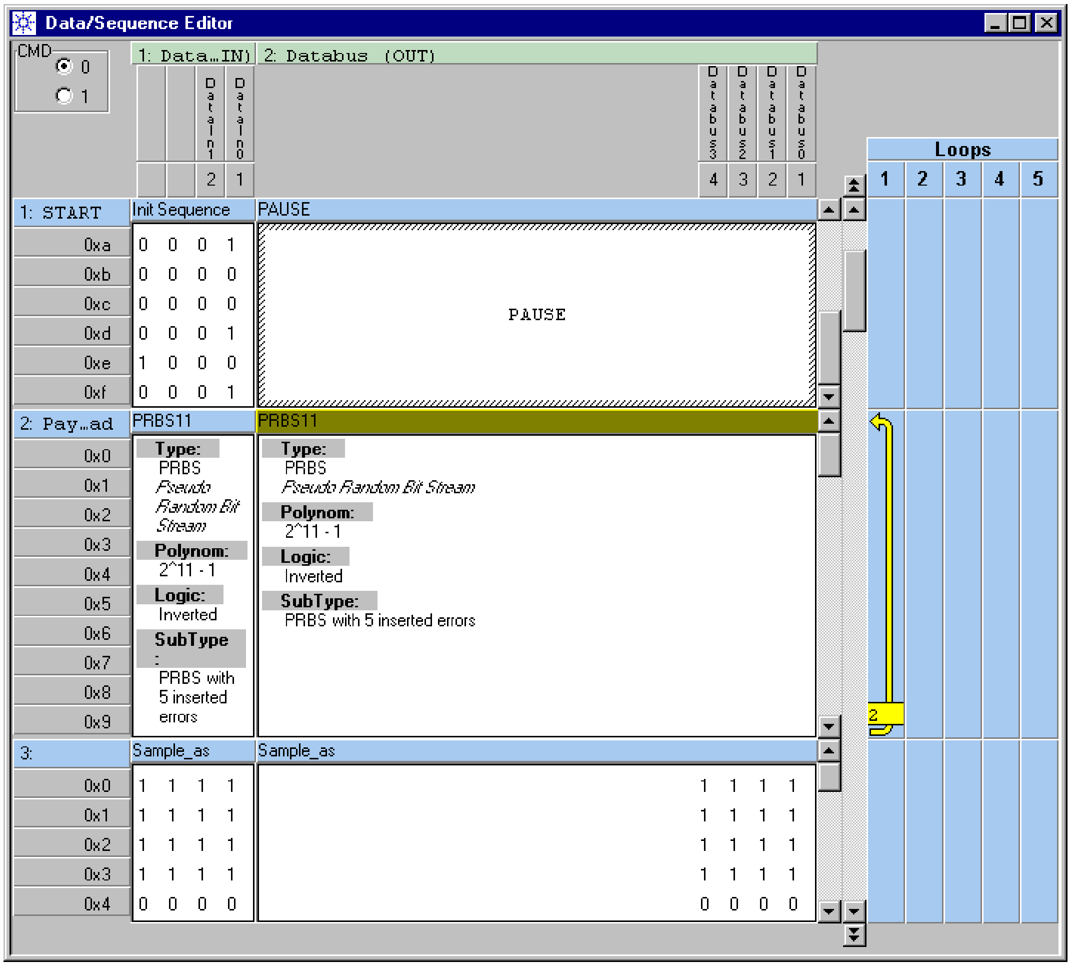
How to Change the Height of a Block

To view more or less blocks:

1 Move the cursor onto the horizontal line that marks the lower block border.

The cursor changes its shape.

2 Hold the mouse button depressed and drag the border line vertically.



1: START — Init Sequence / PAUSE

	Init Sequence				PAUSE
0xa	0	0	0	1	PAUSE
0xb	0	0	0	0	
0xc	0	0	0	0	
0xd	0	0	0	1	
0xe	1	0	0	0	
0xf	0	0	0	1	

2: Pay…ad — PRBS11 / PRBS11

Rows 0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9

PRBS11:
Type: PRBS *Pseudo Random Bit Stream*
Polynom: 2^11 - 1
Logic: Inverted
SubType: PRBS with 5 inserted errors

PRBS11:
Type: PRBS *Pseudo Random Bit Stream*
Polynom: 2^11 - 1
Logic: Inverted
SubType: PRBS with 5 inserted errors

3: — Sample_as / Sample_as

	Sample_as				Sample_as			
0x0	1	1	1	1	1	1	1	1
0x1	1	1	1	1	1	1	1	1
0x2	1	1	1	1	1	1	1	1
0x3	1	1	1	1	1	1	1	1
0x4	0	0	0	0	0	0	0	0

Figure 231 Customized Data / Sequence Editor Window

How to Change the Format of Displayed Addresses

An important difference between the Data/Sequence Editor and the Segment Editor is the meaning of the vector address column at the left-hand side. While the Segment Editor always displays the actual bit vector address in the segment, the Data Sequence Editor displays the clock cycle number at a certain point of time.

If a port uses clock frequency multiplier 1, one vector will be displayed per row (i.e. per cycle number). If another port uses frequency multiplier 4, then four rows of the data segment will occur per cycle for this port. In other words, this port will receive/send four bits for every terminal, while the first one only receives/sends one per terminal.

To change the bit vector/clock cycle number format:

1. Open the context menu of the address column by clicking on it with the right mouse button.
2. Choose from the menu one of the following options:

   Decimal, *Hexadecimal*, or *Octal*.

   Note, that this menu can also be used to jump to any address within the current block by choosing the *Go to* option.

How to Change the Labels of Displayed Traces

To change the view of traces:

1. Open the context menu of the port/terminal display area by clicking on it with the right mouse button.
2. Choose from the menu.

   Choices are: *Trace View, Port View, Terminal View, Channel View* (see *"Data Format" on page 164*).

   Note, that this menu can also be used to insert or delete highlighted traces.

How to Change the Sequence or Edit Segments

The Data/Sequence Editor combines all the functions of the

- Sequence Editor
- Segment Editor

That means, it does not only provide the same capabilities, it also works the same way. In fact, the Data/Sequence Editor just invokes procedures that are already known from the other two editors.

How to Change the Sequence Characteristics

To change the sequence:

1 Open the context menu of a block label by clicking on it with the right mouse button.

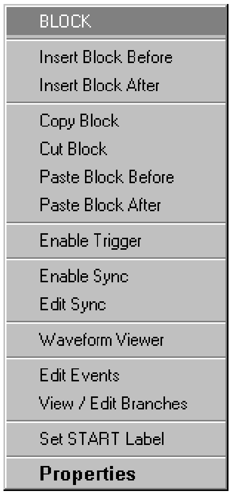


Figure 232 **Block Context Menu**

You have all the options the Sequence Editor provides.

2 Choose the required action from the menu.

For details please refer to *“How to Add, Move or Delete Blocks” on page 288*.

If you wish to change the loops in your sequence, please refer to *“How to Create and Change Loops” on page 296*.

How to Replace a Segment

To replace a segment by another one:

1 Open the context menu of a segment label by clicking on it with the right mouse button.

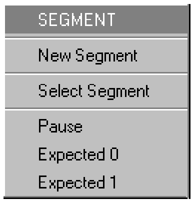


Figure 233 **Segment Context Menu**

You have all the options the Sequence Editor provides. They depend on the selected type of measurement and on the type of port – data input port or data output port.

2 Choose from the menu.

For details please refer to *“How to Replace the Current Segment” on page 292*.

How to Edit the Contents of a Segment

For changing the contents of a segment, you have all the options the Segment Editor provides.

You can change individual bits or highlight rows, columns, or selections and change their contents using the context menu.

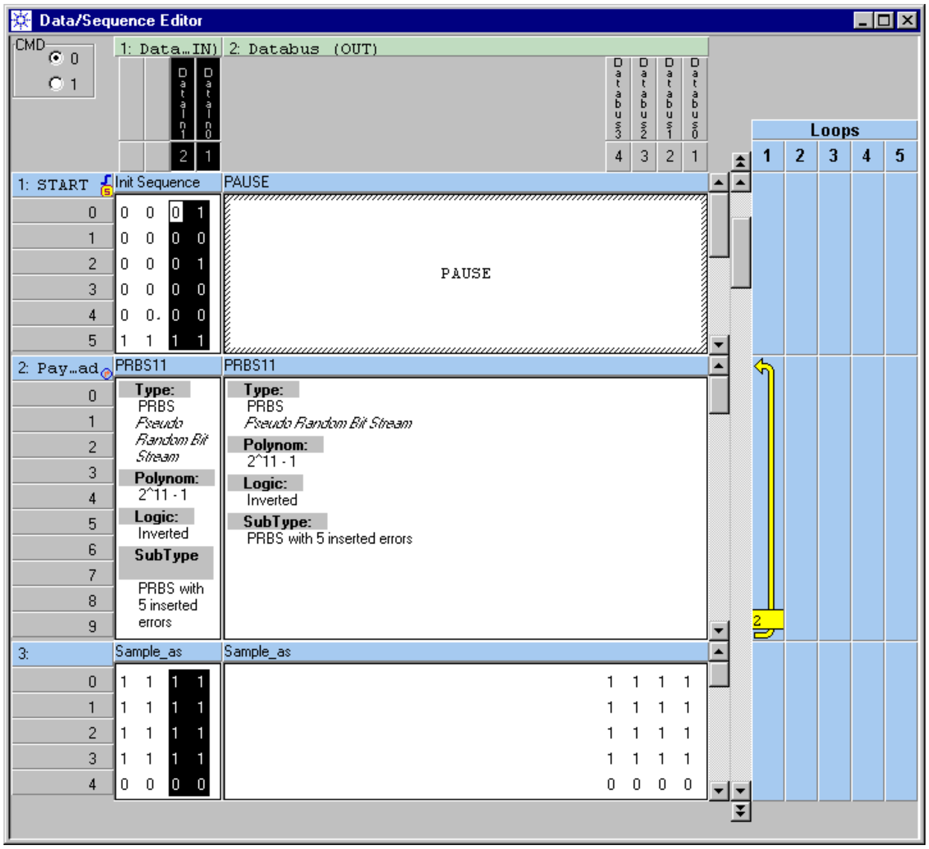


Figure 234 Data / Sequence Editor – Highlighted Column

For details please refer to *"How to Use the Segment Editor's Data Edit Area" on page 317.*

Running the Test

Once the Sequence is complete, you are ready to run the test.

Device Tests can be started and stopped from the user interface. They can also be started and stopped by an external signal applied to the EXT INPUT connector of the master clock module.

NOTE Before starting a test, it is recommended to check the parameter settings of the ports and channels.

To be successful, generated signals must fit to the data formats, voltages and impedances of the DUT. Analyzer sampling point delays must take the signal traveling and processing time of the DUT into account.

This chapter informs you about:

- *"How to Download the Test Sequence" on page 346*
- *"How to View BER Test Results" on page 347*
- *"How to Start/Stop the Test" on page 348*

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo
- In the *Tutorial*, select *"Running the Test and Displaying the Results"*.

How to Download the Test Sequence

To download a sequence to the modules:

1 Click the Prepare button of the tool bar.

Downloading ensures that the test sequence is formally correct and can be executed.

NOTE Downloading is especially important, if you have changed the type of measurement, because pseudo segments like Acquire, Expected 0 and so on apply only to certain tests (see *"How to Replace a Segment by a Pseudo Segment" on page 294*).

Downloading also prepares the Agilent 81250 system for immediate start on a trigger event.

The sequence is also downloaded to the modules when the Run button is pressed. But downloading a complex sequence can take some time. In the meantime there is no output signal generated, nor any input signal captured.

How to View BER Test Results

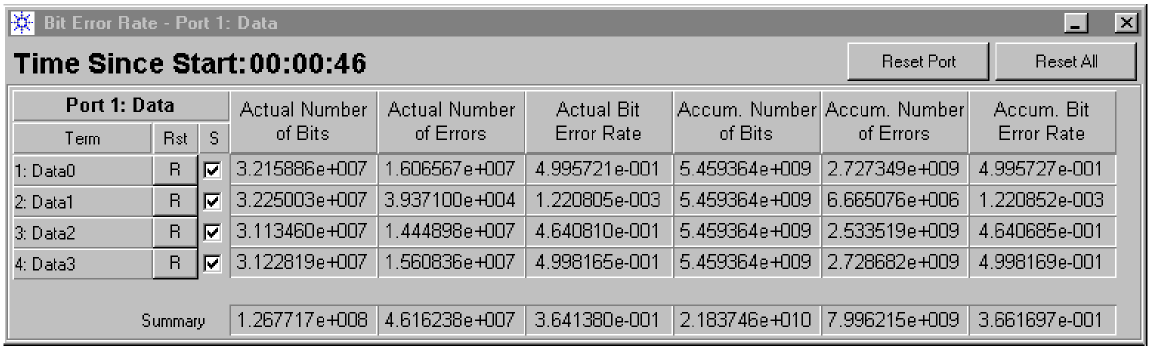
If the test has been set up for measuring the bit error rate:

1 Click the Bit Error Rate Display icon in the tool bar.

This opens the Bit Error Rate window.

2 Drag the right- or left-hand border to view all the columns.

Once the test is running, the window is continually updated. It is therefore recommended to open this window before starting the test.



Bit Error Rate - Port 1: Data

Time Since Start:00:00:46

Port 1: Data			Actual Number of Bits	Actual Number of Errors	Actual Bit Error Rate	Accum. Number of Bits	Accum. Number of Errors	Accum. Bit Error Rate
Term	Rst	S						
1: Data0	R	☑	3.215886e+007	1.606567e+007	4.995721e-001	5.459364e+009	2.727349e+009	4.995727e-001
2: Data1	R	☑	3.225003e+007	3.937100e+004	1.220805e-003	5.459364e+009	6.665076e+006	1.220852e-003
3: Data2	R	☑	3.113460e+007	1.444898e+007	4.640810e-001	5.459364e+009	2.533519e+009	4.640685e-001
4: Data3	R	☑	3.122819e+007	1.560836e+007	4.998165e-001	5.459364e+009	2.728682e+009	4.998169e-001
Summary			1.267717e+008	4.616238e+007	3.641380e-001	2.183746e+010	7.996215e+009	3.661697e-001

Figure 235 Bit Error Rate Display

The resulting BER is shown as actual and accumulated values per terminal and port. The elapsed time since start of the measurement is also displayed.

All counters can be reset at any time, either individually per terminal (*R* buttons in the *Rst* column), or per port (*Reset Port* button) or all at once (*Reset All* button).

There is a summary line at the bottom. By clicking the marker in the *S* column, terminals can be excluded from or included in the summary line.

TIP The sequence of the columns can be customized. To move a column to a different position, click the column header with the left mouse button and drag the column horizontally to the desired position.

How to Start / Stop the Test

To start the test:

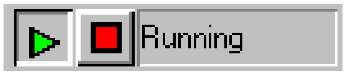
1 Ensure that the frontends are connected to the DUT.

The Connectors On/Off button can be used to disconnect all frontends (by switching relays inside the frontends) and to re-establish the previously specified connections.

2 Click the Run button.

If the test has been set up to be controlled by an external start trigger, the user interface will display HALTED and the system will wait for that trigger. If not, it starts immediately.

The test will run until the test sequence is completely executed or the capture memory is full or, if it is controlled by an external stop trigger, until the trigger is set—whichever comes first.



If the test sequence includes an infinite loop, stop the test by clicking the Stop button.

Clicking the Stop button is also required, if the status *Finished* is displayed.

Viewing Generated and Captured Data

After running one of the tests

- *Capture Data*
- *Compare and Acquire around Error*
- *Compare and Capture*

you can review the captured data.

After running one of the tests *Compare and Acquire around Error* or *Compare and Capture* you can also investigate errors, the pattern around errors, and generated data.

The results of a bit error rate measurement are displayed in the Bit Error Rate window (see *"How to View BER Test Results" on page 347*).

This chapter informs you about:

- *"How to View Captured Test Results" on page 350*
- *"How to View Waveforms" on page 356*

How to View Captured Test Results

Captured data as well as errors can be visually checked in the Error State Display.

How to Start the Error State Display

To open the Error State Display:

1 Click the Error State Display icon of the tool bar.

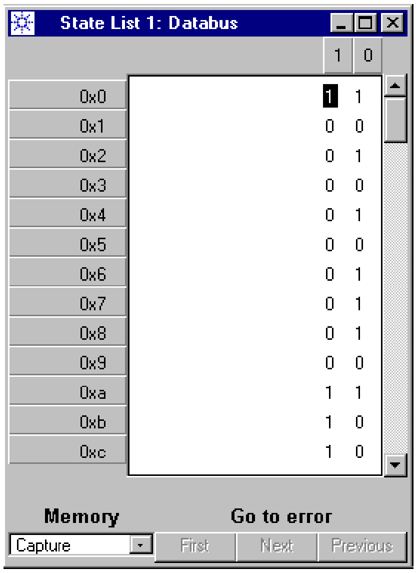
Alternatively, you can also open the *View* menu and choose *Result Displays*.

The Error State Display identifies the DUT output port and shows the memory contents of the analyzer channels.

How to Operate the Error State Display

The Error State Display has three display modes. It can show:

- Captured Data: Shows what has been captured.
- Compared Data: Shows captured data where errors are highlighted.
- Error Data: Shows errors only.



State List 1: Databus

	1	0
0x0	1	1
0x1	0	0
0x2	0	1
0x3	0	0
0x4	0	1
0x5	0	0
0x6	0	1
0x7	0	1
0x8	0	1
0x9	0	0
0xa	1	1
0xb	1	0
0xc	1	0

Memory: Capture

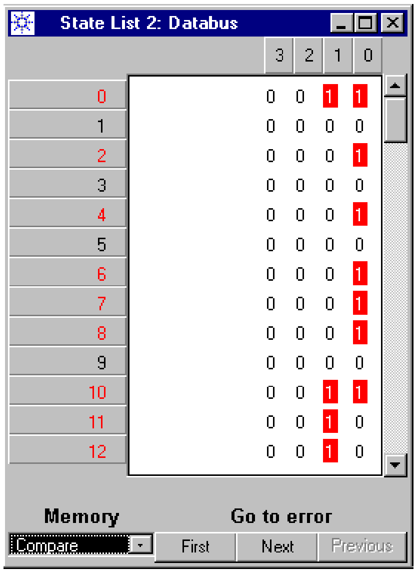
Figure 236 Error State Display in Capture Mode

The window has two active areas with context menus—the address column and the data display box.

In the address column, you can change the address display format (choices are *Decimal*, *Hexadecimal*, or *Octal*) or choose the *Go to* option to specify the start address of the display.

In the data display box, you can change the data display format. Choices are *Binary*, *Hexadecimal*, or *Octal*.

In binary mode, each trace has its own column. In hexadecimal mode, four traces are combined in one column (range 0 to F_{hex}). In octal mode, three traces are combined in one column (range 0 to 7_{oct}).



	3	2	1	0
0	0	0	1	1
1	0	0	0	0
2	0	0	0	1
3	0	0	0	0
4	0	0	0	1
5	0	0	0	0
6	0	0	0	1
7	0	0	0	1
8	0	0	0	1
9	0	0	0	0
10	0	0	1	1
11	0	0	1	0
12	0	0	1	0

Figure 237 Error State Display in Compare Mode

In Compare mode, the window provides three Go-to-Error buttons to move quickly from one error to the next or previous.

How to Transfer Captured Data Into a Segment

Captured data can be saved in a data segment.

This makes it possible to use the response of a device as a reference for future devices of the same kind. If the segment with the captured data is used to specify the expected data for tests to come, the system precisely measures all deviations from the gold standard.

There are two ways to transfer captured data into a segment. You can:

- Save the captured data as a new segment
- Copy the captured data to the clipboard and then paste that data into a segment

How to Save Captured Data as a New Segment

To convert captured data to a new segment:

1 When the Error State Display is active, open the *File* menu and choose *Save Segment As*.

2 Enter the new segment’s file name.

3 Click OK.

NOTE Before using the segment for interpreting received data, check and, if necessary, change the state coding (see *“How to Create a Memory Segment” on page 312*).

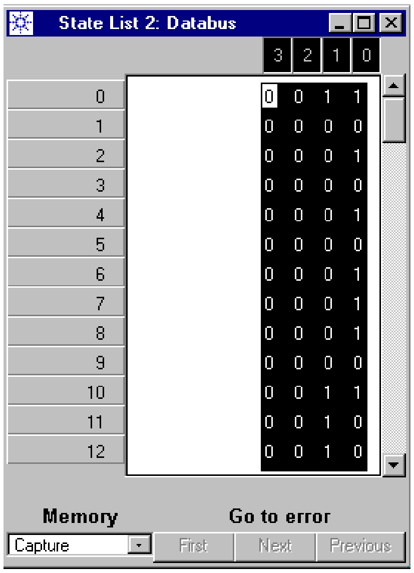
How to Copy Captured Data Into a Segment

To copy captured data into a new or existing segment:

1 Put the Error State Display in Capture mode.

2 Drag the cursor across the trace numbers.

This highlights all the captured data. You can of course also highlight a data section.



	3	2	1	0
0	0	0	1	1
1	0	0	0	0
2	0	0	0	1
3	0	0	0	0
4	0	0	0	1
5	0	0	0	0
6	0	0	0	1
7	0	0	0	1
8	0	0	0	1
9	0	0	0	0
10	0	0	1	1
11	0	0	1	0
12	0	0	1	0

Figure 238 Highlighted Captured Data

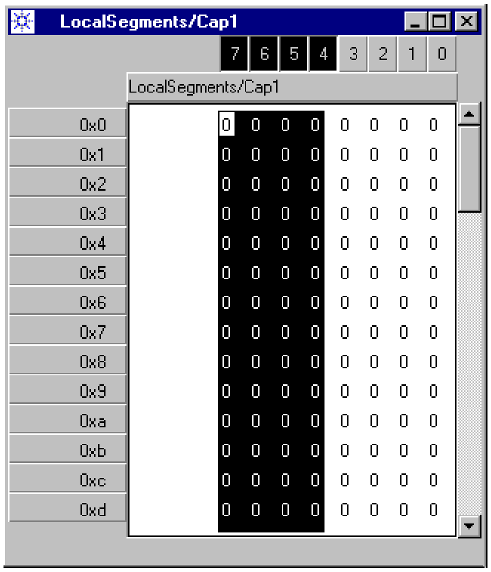
3 Open the context menu and choose *Copy*.

This copies the data to the clipboard.

4 Create a new memory segment (see *"How to Create a New Segment" on page 310*).

When creating the segment, ensure that it is long and wide enough to hold the pattern you wish to include.

5 In the Segment Editor, highlight the traces that shall get the data pattern.



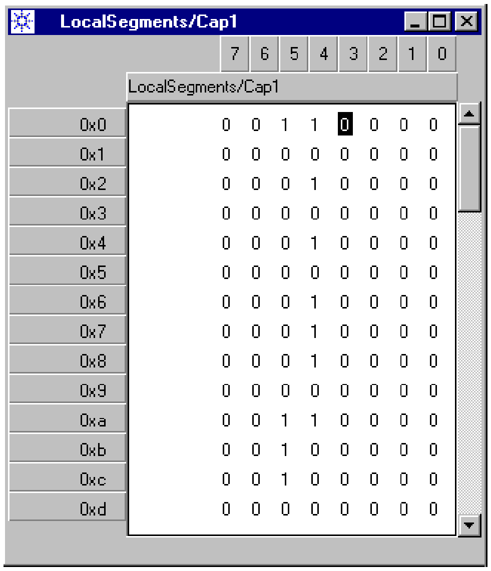
LocalSegments/Cap1

	7	6	5	4	3	2	1	0
0x0	0	0	0	0	0	0	0	0
0x1	0	0	0	0	0	0	0	0
0x2	0	0	0	0	0	0	0	0
0x3	0	0	0	0	0	0	0	0
0x4	0	0	0	0	0	0	0	0
0x5	0	0	0	0	0	0	0	0
0x6	0	0	0	0	0	0	0	0
0x7	0	0	0	0	0	0	0	0
0x8	0	0	0	0	0	0	0	0
0x9	0	0	0	0	0	0	0	0
0xa	0	0	0	0	0	0	0	0
0xb	0	0	0	0	0	0	0	0
0xc	0	0	0	0	0	0	0	0
0xd	0	0	0	0	0	0	0	0

Figure 239 Empty Segment

6 Open the context menu and choose *Paste*.

The result is shown below:



LocalSegments/Cap1	7	6	5	4	3	2	1	0
0x0	0	0	1	1	0	0	0	0
0x1	0	0	0	0	0	0	0	0
0x2	0	0	0	1	0	0	0	0
0x3	0	0	0	0	0	0	0	0
0x4	0	0	0	1	0	0	0	0
0x5	0	0	0	0	0	0	0	0
0x6	0	0	0	1	0	0	0	0
0x7	0	0	0	1	0	0	0	0
0x8	0	0	0	1	0	0	0	0
0x9	0	0	0	0	0	0	0	0
0xa	0	0	1	1	0	0	0	0
0xb	0	0	1	0	0	0	0	0
0xc	0	0	1	0	0	0	0	0
0xd	0	0	0	0	0	0	0	0

Figure 240 **New Segment**

Copied data that does not fit into the segment is ignored.

7 Save the segment.

How to View Waveforms

Generated, expected and captured data can be displayed in graphical form with the Waveform Viewer.

How to Start the Waveform Viewer

To open the Waveform Viewer:

1 Click the Waveform Viewer icon of the tool bar.

Alternatively, you can also open the View menu and choose *Result Displays*.

The Waveform Viewer identifies the test sequence and the ports:

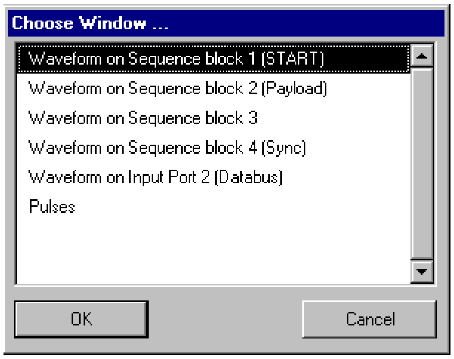

Choose Window ...
Waveform on Sequence block 1 (START)
Waveform on Sequence block 2 (Payload)
Waveform on Sequence block 3
Waveform on Sequence block 4 (Sync)
Waveform on Input Port 2 (Databus)
Pulses
OK
Cancel

Figure 241 Waveform Viewer Selection Window

If you select a data block of the test sequence or a pulse port, generated and expected data will be displayed in **time mode**. The available resolution is the segment resolution, the unit is nanoseconds (ns). This enables you to check the delays that have been set up. Note that PRBS/PRWS data cannot be displayed in time mode.

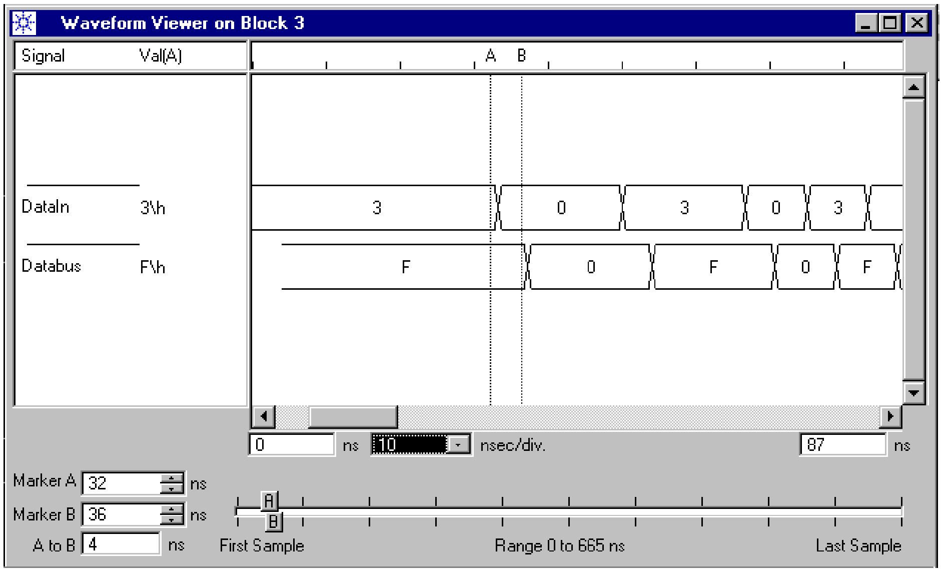
If you select a DUT output port (= analyzer input), the data will be displayed in **sample mode**. This corresponds to the way the data has been acquired. The total number of samples is the number of captured vectors times the maximum factor of the FMR (see *“Frequency Multiplier and Segment Resolution” on page 68*).

2 Choose from the menu.

Description of the Waveform Viewer Display

The Waveform Viewer comes up with a default configuration which can be changed at will.

The figure below shows an example of a block which includes one data input port (DataIn) and one data output port (Databus).



Waveform Viewer on Block 3

Signal	Val(A)
DataIn	3\h
Databus	F\h

0 ns 10 nsec/div. 87 ns

Marker A 32 ns

Marker B 36 ns

A to B 4 ns

Figure 242 Waveform Viewer in Time Mode

Two traces show the code transitions and the generated and expected codes in hexadecimal format.

The display provides two markers, A and B. Their current position and distance is indicated in the lower left-hand corner. They can be moved with the verniers or by dragging their handles along the ruler.

The column *Val(A)* shows the codes at the position of marker A. The literal \h indicates that these are hex codes.

The current resolution is 10 ns/div but can be changed.

How to Operate the Waveform Viewer

The context menu provides the following options:

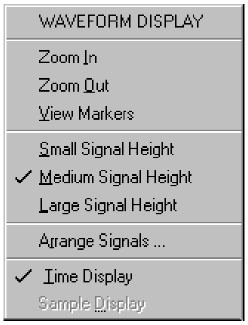


Figure 243 Waveform Viewer Context Menu

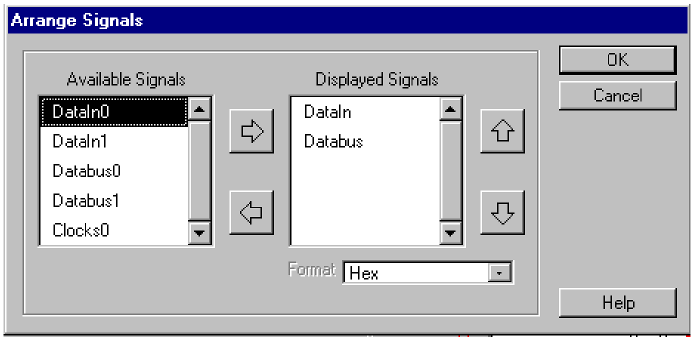
You can:

- Zoom in, zoom out, or view the area between the markers.
- Increase or decrease the waveform amplitudes.
- Rearrange the display.

To view additional or different data:

1 Click *Arrange Signals*.

The Arrange Signals Window appears.



Format Hex

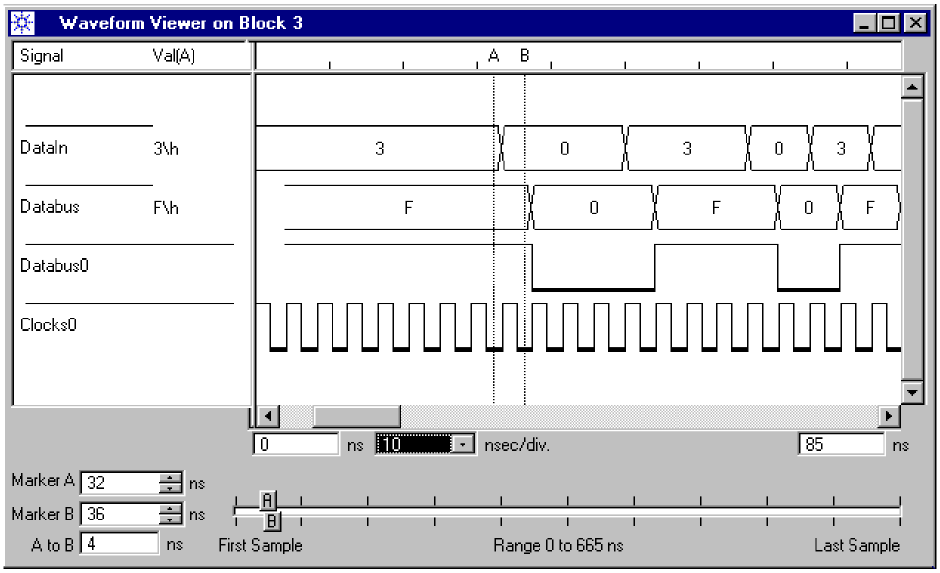
Figure 244 Waveform Viewer Arrange Signals Window

2 To view an item in the display, highlight it in the list of *Available Signals* and click the right-arrow.

3 To remove an item from the display, highlight it in the list of *Displayed Signals* and click the left-arrow.

4 To move an item in the list of *Displayed Signals*, highlight it and click the up- or down-arrow.

The result may look as shown below:



Waveform Viewer on Block 3

Signal	Val(A)
DataIn	3\h
Databus	F\h
Databus0	
Clocks0	

0 ns 10 nsec/div. 85 ns

Marker A 32 ns

Marker B 36 ns

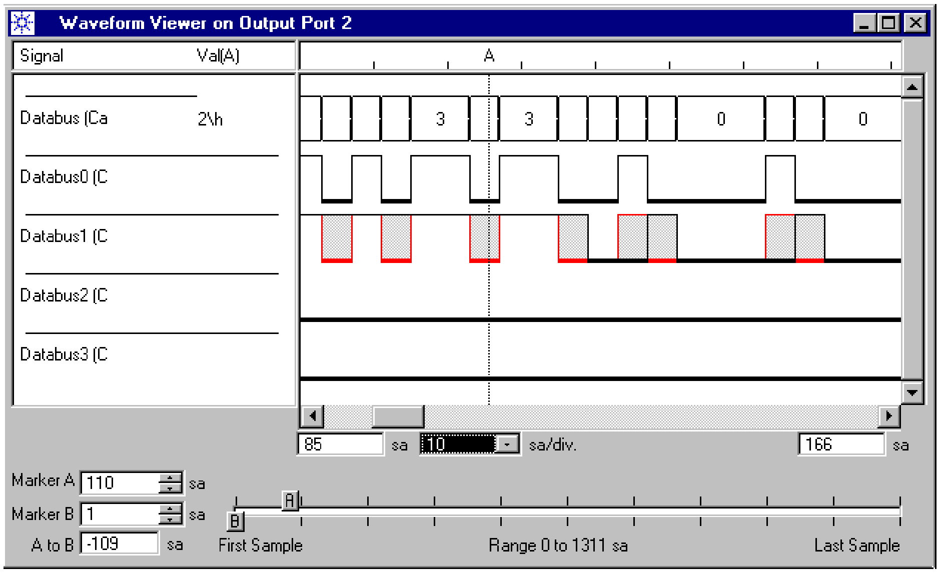
A to B 4 ns

First Sample Range 0 to 665 ns Last Sample

Figure 245 Waveform Viewer in Time Mode—Additional Waveforms

NOTE If you are viewing data input or pulse terminals and have sourced an added channel (see also *“How to Combine Generator Channels” on page 251*): The Waveform Viewer shows only the signal of the channel that has a connector.

If you have opened the Waveform Viewer for a data output port, it shows the summary of the captured data and the individual channels in **sample mode**. If error recognition was enabled, it shows also the deviations from expected data.



Databus (Ca 2\h
Databus0 (C
Databus1 (C
Databus2 (C
Databus3 (C

85 sa 10 sa/div. 166 sa

Marker A 110 sa
Marker B 1 sa
A to B -109 sa

Figure 246 Waveform Viewer—Waveforms in Sample Mode

The signals are identified as

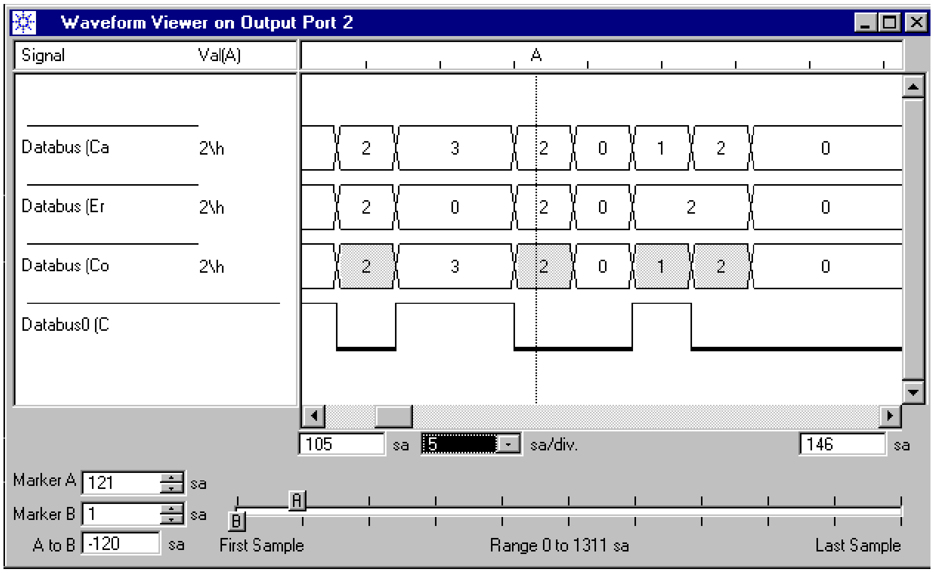
- (Capt)—captured data,
- (Comp)—compared data,
- (Err)—error data.

The display is presently limited to 10 characters.

A zero line tells you that nothing was received or expected.

The *Arrange Signals* menu now offers additional options: You can select between captured, compared, and error data, and, thus, compose an individual display.

The result may look as shown below:



Waveform Viewer on Output Port 2

Signal	Val(A)
Databus (Ca	2\h
Databus (Er	2\h
Databus (Co	2\h
Databus0 (C	

105 sa 5 sa/div. 146 sa

Marker A 121 sa

Marker B 1 sa

A to B -120 sa

Figure 247 Waveform Viewer in Sample Mode—Additional Waveforms

Using Auxiliary Functions

This chapter provides information on auxiliary functions which are not directly related to test setup and execution. The following topics are covered:

- *"How to Compensate for Internal and External Delays" on page 364*—the description of the Deskew Editor
- *"How to Deskew Optical Connections" on page 374*—a guideline on how to align the phases of multiple optical signals
- *"How to Export/Import Settings or Segments" on page 378*—an explanation of the export and import functions
- *"How to Execute Firmware Commands" on page 382*—a nice feature that allows you to send firmware commands to the hardware and check the responses
- *"How to Use the System Starter Utilities" on page 385*—the description of two tools for controlling tests on several systems in parallel

How to Compensate for Internal and External Delays

Precise measurements require exact timing. Generated signals must reach the DUT simultaneously, response signals must be captured at the same point of time by all analyzers.

The Agilent 81250 Parallel Bit Error Ratio Tester supports timing adjustments at the generator/analyzer connectors of the system, at the input and output connectors of the DUT board, and even at the pins of the DUT.

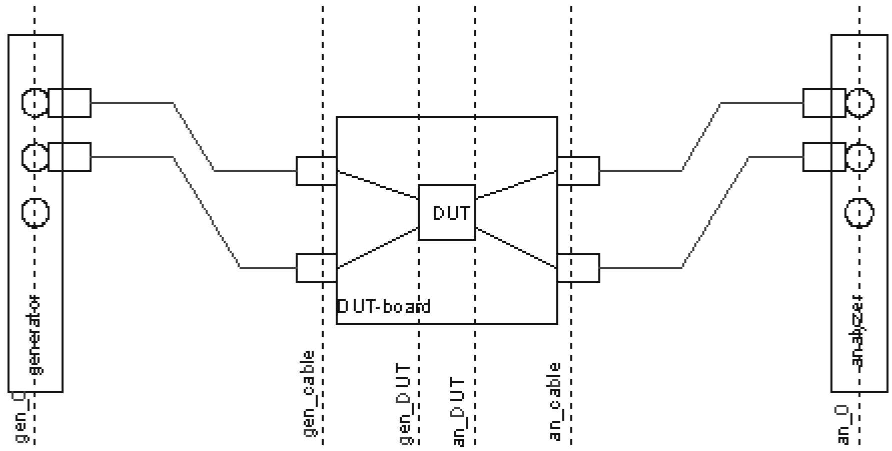


Figure 248 **Supported Timing Adjustments**

For this purpose, the ParBERT software provides three procedures:

Zero adjust After installing new modules or after replacement of frontends a *zero adjust* procedure has to be performed to synchronize the new generators or analyzers with the ones already installed.

Cable delay To compensate for signal delays in the used cables, a *cable delay compensation* procedure can be performed.

Cable and propagation delay To compensate for both propagation delays on the DUT board and delays in the cables, a *cable and propagation delay compensation* procedure can be performed.

NOTE If any frontends or modules have been replaced or added to the system, you must first run *Delay Auto Calibration* from the *System* menu. This has to be done **before** the zero adjust and cable delay compensation procedures are performed. For details see *"Delay Auto Calibration" on page 170*.

Delay compensation is performed with the Deskew Editor.

NOTE If you need to deskew a ParBERT 43G system because modules or frontends have been exchanged, some restrictions apply. See *"Additional Characteristics of ParBERT 43G Systems" on page 116*.

Multi-Media Guided Tour, Tutorial and Getting Started

As an additional source of information, the Multi-Media Guided Tour, Tutorial and Getting Started provide a comprehensive overview of the Agilent 81250 Parallel Bit Error Ratio Tester.

If it is installed on your system, you will find it in the Windows start menu under *Programs – Agilent 81250 Tutorial.*

If not, you can download it from the web through

- http://www.agilent.com/find/81250demo

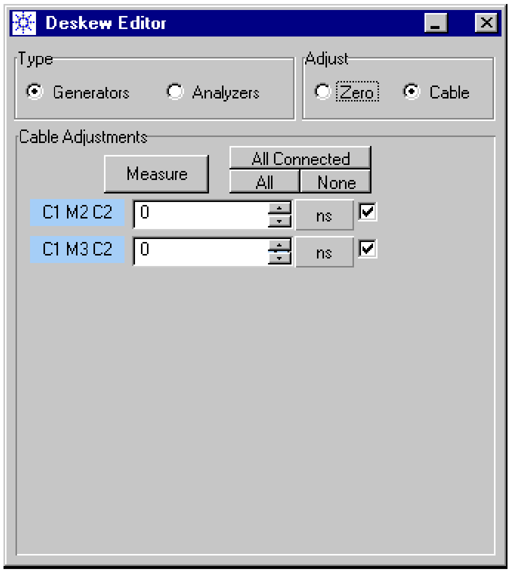
♦ In the *Tutorial*, select *"Establishing Uniform Signal Delays"*.

How to Start the Deskew Editor

To start the Deskew Editor:

1 Open the *Go* menu.

2 Click *Deskew Editor*.



Type: ● Generators ○ Analyzers

Adjust: ○ Zero ● Cable

Cable Adjustments

C1 M2 C2 0 ns ☑

C1 M3 C2 0 ns ☑

Figure 249 Deskew Editor—Start Window

By default, the Deskew Editor identifies all the installed generator channels and suggests a cable delay measurement.

The identification of a generator or analyzer channel is:

Cx My Cz (ClockgroupNumber – ModuleNumber – ConnectorNumber), such as C1 M3 C4. See also *"Identification of Hardware Resources" on page 56*.

How to Adjust the Instrument Connectors

Zero adjustment ensures that an edge produced by the generator channels appears simultaneously at all generator connectors, and that received data is sampled at all analyzer connectors at the same point of time.

Prerequisites for Zero Adjustment

To perform this procedure, you need a reference SMA cable with known signal delay.

Zero Adjustment Procedure

Once the Deskew Editor has been started:

1 Decide whether you wish to deskew generator or analyzer channels.

If you want to deskew all the installed channels, start with the generators.

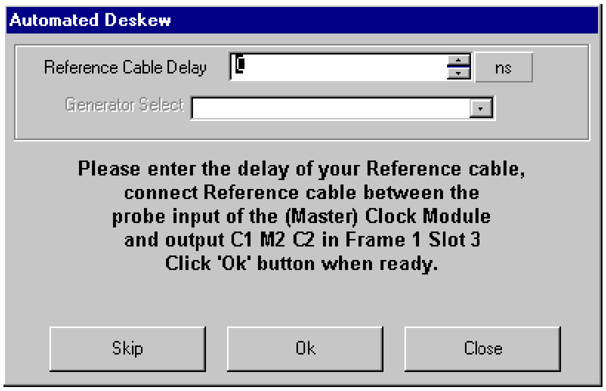
2 Click the *Zero* button in the *Adjust* field to select zero adjust.

The last measured delay values are displayed.

By default, all connectors are marked for the zero adjust. With the *None* button you can deselect all and then mark just the new connectors for the zero adjust procedure.

3 Click the *Measure* button.

For generators, the following window pops up:



Automated Deskew

Reference Cable Delay: 0 ns

Generator Select:

Please enter the delay of your Reference cable, connect Reference cable between the probe input of the (Master) Clock Module and output C1 M2 C2 in Frame 1 Slot 3 Click 'Ok' button when ready.

Figure 250 Deskew Editor—Zero Adjust Window for a Generator Channel

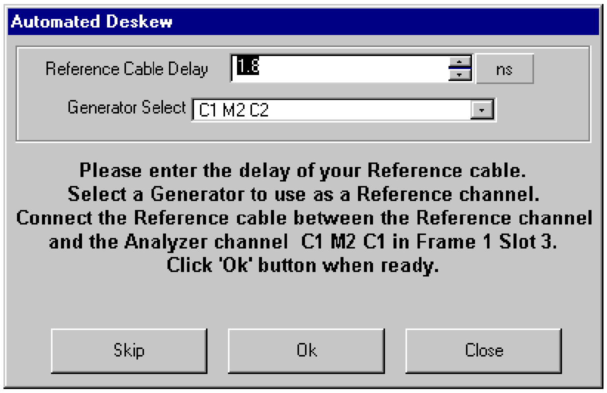
4 Follow the instructions given in this window.

Enter the reference cable delay in the provided field. The typical delay of a 1 m (3 feet) SMA cable is about 3.5 ns.

The instructions will guide you from one connector to the next.

5 Adjust also the analyzers.

The zero adjustment of analyzers requires a signal generator to be used as a reference channel. The system proposes the first of the installed generator channels. If there is no generator frontend installed, the TRIGGER OUTPUT of the clock module is suggested.



Reference Cable Delay 1.8 ns

Generator Select C1 M2 C2

Please enter the delay of your Reference cable.
Select a Generator to use as a Reference channel.
Connect the Reference cable between the Reference channel and the Analyzer channel C1 M2 C1 in Frame 1 Slot 3.
Click 'Ok' button when ready.

Figure 251 Deskew Editor—Zero Adjust Window for an Analyzer Channel

The instructions guide you from one connector to the next. Finally, the measured zero adjust values are automatically entered into the initial table.

How to Compensate for Cable Delays

The cable delay compensation for the generators assures that the edges of all generator outputs appear at the same time at the end of the cables used in the setup.

The cable delay compensation for the analyzers assures that all output signals of the DUT are sampled at the same time at the end of the cables, close to the DUT.

Prerequisites for Cable Delay Compensation

To perform this procedure, you need all the cables which are going to be used by your application of the system.

The cable delay compensation range is ±23 ns.

Perform the cable delay compensation first for the generator outputs, then for the analyzer inputs.

It is recommended to use cables of equal type and length for all inputs and outputs.

Cable Delay Compensation Procedure

To compensate for cable delays:

1 Connect the cables you will use in your application to the system's generator connectors.

2 Create an image of your application in the Connection Editor. Group the signals to ports and make the connections in the scheme.

In this example, the following setup is used:

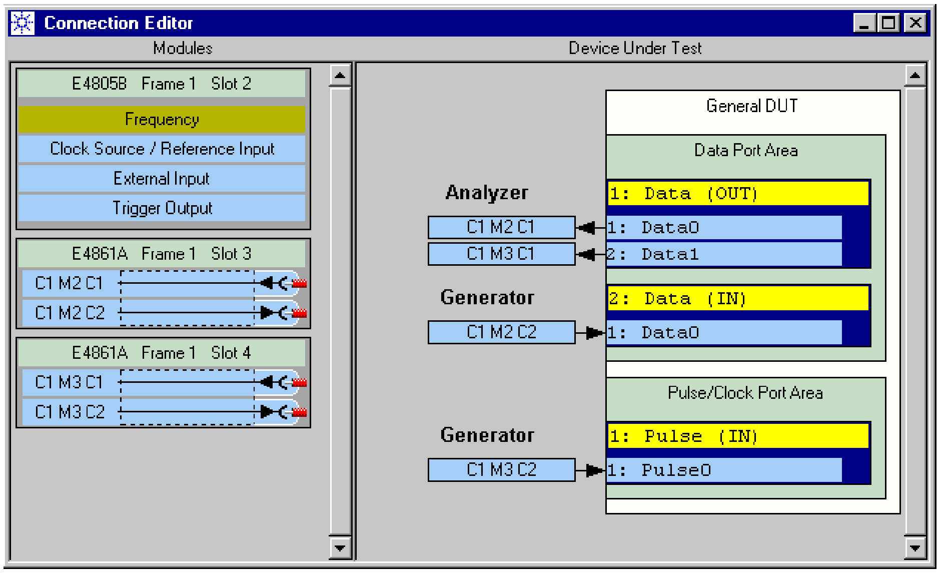


Figure 252 Connection Editor—Deskew Example Setup

3 Use the Parameter Editor to set the levels for all DUT input/output ports.

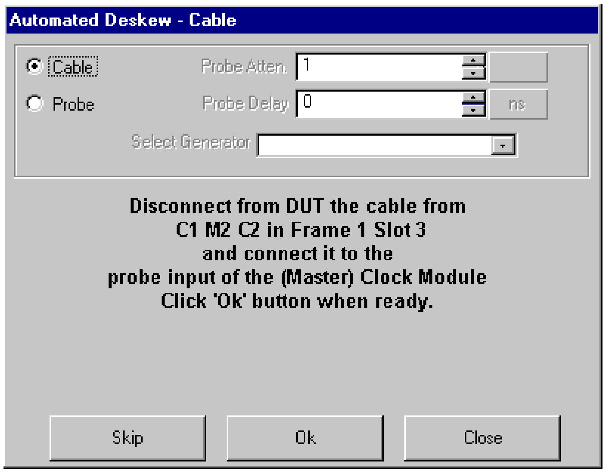
4 Open the Deskew Editor (see *"How to Start the Deskew Editor" on page 366*).

By default, generators are selected in the *Type* field and cable delay measurement is enabled.

If you have connected the generator outputs in the Connection Editor, then you can easily mark the generators for this measurement by clicking on *All connected*.

5 Click the *Measure* button.

The following window pops up.



Automated Deskew - Cable

● Cable
○ Probe

Probe Atten. 1

Probe Delay 0 ns

Select Generator

Disconnect from DUT the cable from
C1 M2 C2 in Frame 1 Slot 3
and connect it to the
probe input of the (Master) Clock Module
Click 'Ok' button when ready.

Figure 253 Deskew Editor—Cable Measurement Window for a Generator Channel

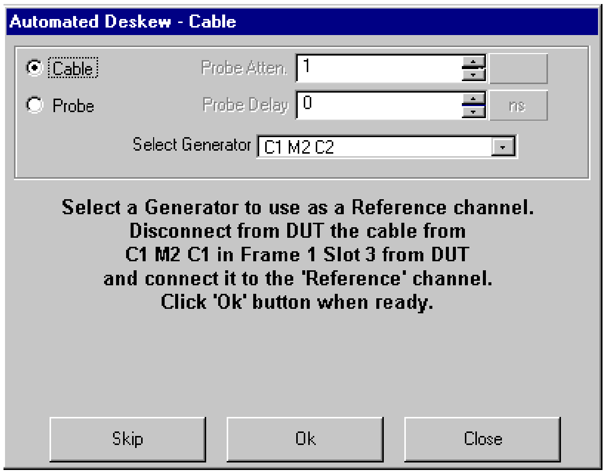
6 Follow the instructions given in this window.

The measured cable delay value is automatically entered into the table and the instruction to connect the next generator output is displayed.

7 Continue with the analyzer cables.

This is essentially the same procedure, except that you do not use the PROBE connector of the clock module, but one of the generator channels as a reference. The system proposes the first of the installed generator channels.

In our example, this is C1 M2 C2.



Automated Deskew - Cable

● Cable
○ Probe

Probe Atten. 1

Probe Delay 0 ns

Select Generator C1 M2 C2

Select a Generator to use as a Reference channel.
Disconnect from DUT the cable from
C1 M2 C1 in Frame 1 Slot 3 from DUT
and connect it to the 'Reference' channel.
Click 'Ok' button when ready.

Figure 254 Deskew Editor—Cable Measurement Window for an Analyzer Channel

How to Compensate for Cable and DUT Board Delays

Performing a cable delay and propagation delay compensation assures that the edges of all generator outputs of the Agilent 81250 Parallel Bit Error Ratio Tester are applied to the DUT input pins at the same time. The procedure also assures that all output signals of the DUT are sampled at the same point of time at the DUT output pins.

Prerequisites for Cable and DUT Board Delay Compensation

This procedure uses the PROBE input of the master clock module.

To perform this procedure, you need all the cables which are going to be used. It is recommended to use cables of the same type and same length at all inputs and outputs. You need also an active probe, e.g. the Agilent 1144A, 800 MHz Active Probe, 10:1.

Cable and DUT Board Delay Compensation Procedure

1 To compensate for cable and propagation delays:

1 Connect the cables you will use in your application to the Agilent 81200 system’s output connectors.

2 Create an image of your application in the Connection Editor. Group the signals to ports and make the connections in the scheme.

3 In the Parameter Editor set the levels for all DUT input and output ports.

4 Open the Deskew Editor (see *“How to Start the Deskew Editor” on page 366*).

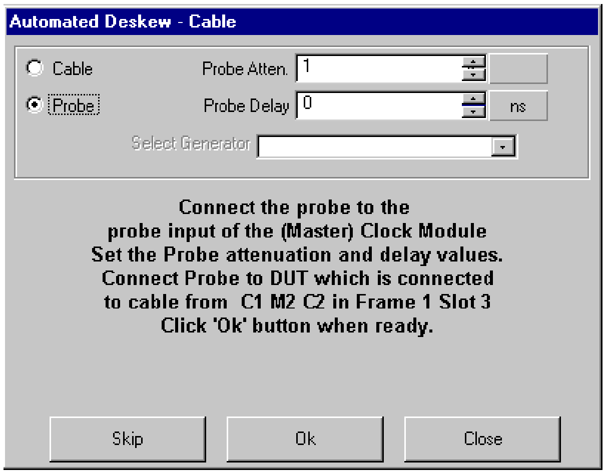
Per default, generators are selected in the *Type* field and cable delay measurement is enabled.

If you have connected the generator outputs in the Connection Editor, then you can easily mark the outputs for this measurement by clicking on *All connected*.

5 Click the *Measure* button.

6 Activate *Probe*.

The following window pops up:



Automated Deskew - Cable

○ Cable Probe Atten. 1

● Probe Probe Delay 0 ns

Select Generator

Connect the probe to the
probe input of the (Master) Clock Module
Set the Probe attenuation and delay values.
Connect Probe to DUT which is connected
to cable from C1 M2 C2 in Frame 1 Slot 3
Click 'Ok' button when ready.

Figure 255 Deskew Editor—Probe Measurement Window for a Generator Channel

7 Enter the *Probe Attenuation* factor

For the Agilent 1144A Active Probe this is 10.

8 Enter the *Probe Delay*.

If you don’t know the probe delay, you can make a first measurement by connecting the probe directly to the first generator output. Start the measurement by clicking OK. The result is the probe’s propagation delay.

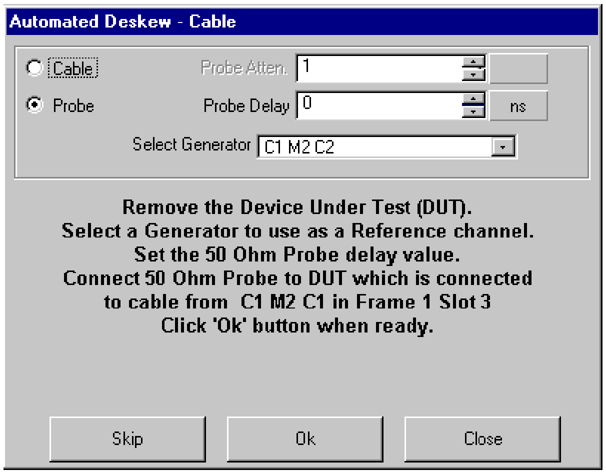
Close the Automated Deskew window. Click *Measure* again. Enter the resulting value as the probe delay and repeat the measurement. Now the resulting value for this output should be 0 ns.

9 Follow the instructions given in the window.

The measured delay value is automatically entered into the table, and the instruction to connect the next generator output is displayed.

10 Continue with the analyzers.

Here, you connect the analyzer inputs with one of the generator channels. This is the reference. Therefore, the generators have to be aligned first. The system proposes the first of the installed generator channels.



Probe Atten. 1
Probe Delay 0 ns
Select Generator C1 M2 C2

Remove the Device Under Test (DUT).
Select a Generator to use as a Reference channel.
Set the 50 Ohm Probe delay value.
Connect 50 Ohm Probe to DUT which is connected
to cable from C1 M2 C1 in Frame 1 Slot 3
Click 'Ok' button when ready.

Figure 256 Deskew Editor—Probe Measurement Window for an Analyzer Channel

For analyzers, the “probes” are the 50 Ohm cables that fit to the DUT board and are going to be used for capturing the DUT output signals. The *Probe Attenuation* factor of these cables is 1. The *Probe Delay* depends on the cable length.

How to Deskew Optical Connections

Like electrical connections, optical connections have to be deskewed. Deskewing ensures that all generated optical signals arrive at the same point of time at the DUT and all received optical signals arrive at the same point of time at the analyzers.

For optical-electrical generators, the “cable delay” includes the delays caused by the cable to the E/O converter, the E/O converter itself, and the optical fiber.

For optical-electrical analyzers, the “cable delay” includes the delays caused by the optical fiber, the O/E converter, and the cables to the electrical analyzer.

Prerequisites for Deskewing Optical Connections

Generators and analyzers must be enabled (watch the green LEDs).

For measuring the cable delays, you need an optical storage oscilloscope with at least two channels. This oscilloscope must allow you to compare the timing of electrical and optical signals.

Zero Adjustment

If any frontends or modules have been replaced or added to the system, you have to perform the *Delay Auto Calibration* (see *“Delay Auto Calibration” on page 170*).

After that, you perform the zero adjustment. This is done in electrical mode.

If the clock module was replaced, zero adjustment has to be performed for all generator and analyzer channels. If channels have been added, zero adjustment has to be performed for these channels.

For optical modules/channels, proceed as follows:

1. Put optical-electrical generators/analyzers into electrical mode (see *"How to Set the Mode of an Optical-Electrical Connector" on page 209*).
2. Use the Deskew Editor and a reference cable to align the electrical connectors. Perform the zero-adjust procedure (see *"How to Adjust the Instrument Connectors" on page 366*).

Cable Deskew

Cable delay compensation is required if any of the glass fibers used for connecting to the DUT has been changed. It is also required if you load or create a setting without importing known cable delay values.

For optical modules/channels, proceed as follows:

1. Put optical-electrical generators/analyzers into optical mode (see *"How to Set the Mode of an Optical-Electrical Connector" on page 209*).
2. Decide on the timing reference.

   You can use one of the optical generator channels as a timing reference.

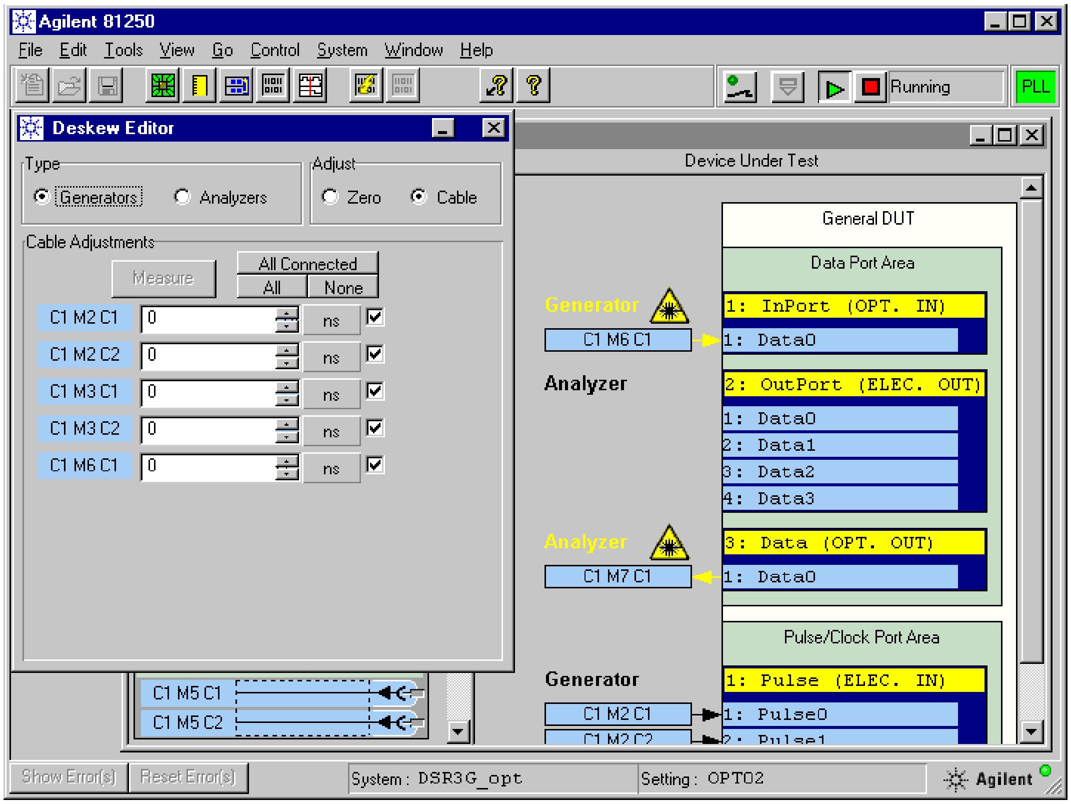
   If desired, you can also use an electrical signal as a reference, for example a clock pulse.
3. Prepare the system. Set up the frequency, connections, and sequence that will be used for the measurement you are planning.

Deskewing Optical Generators

To determine and set the "cable delay" of optical generators:

1. Connect the oscilloscope to the timing reference and to the first optical channel (fiber end) to be deskewed.
2. Click the *Run* button.
3. Set the Deskew Editor to *Adjust Cable*.

TIP The Deskew Editor shows all channels. It may be a good idea to also have the Connection Editor window at hand. The Connection Editor makes it easy to distinguish between electrical and optical channels.



Deskew Editor

Type: ● Generators ○ Analyzers

Adjust: ○ Zero ● Cable

Cable Adjustments

C1 M2 C1	0	ns	☑
C1 M2 C2	0	ns	☑
C1 M3 C1	0	ns	☑
C1 M3 C2	0	ns	☑
C1 M6 C1	0	ns	☑

System : DSR3G_opt

Setting : OPT02

Figure 257 Deskew Editor—Generator Cable Delay Adjustment

In the figure above, the optical generator is identified as C1 M6 C1.

4 With the oscilloscope, measure the delay between the reference and the optical signal.

5 Enter the measured delay in the Deskew Editor window.

6 Repeat the steps 4 and 5 for all optical fibers.

Deskewing Optical Analyzers

If your system includes an optical generator:

1 Use the automatic cable delay compensation procedure (see *"How to Compensate for Cable Delays" on page 368*).

2 Connect the analyzers one after the other with the generator.

NOTE The O/E converter has an inherent delay (see the technical specifications). On the other hand, the measurement range of the automatic cable delay compensation procedure is limited to approximately ±23 ns.

If you are using fibers that are longer than 1.6 m (corresponding to a signal delay of about 8 ns), the total delay may exceed the capability of the automatic cable delay compensation.

If this is the case, you have to measure the delay and enter the value manually (see below).

If your system does not have any optical generator:

1 Connect a laser source to the analyzer.

2 Using an oscilloscope, measure the delay between the optical source and the electrical analyzer input connector.

3 Enter the result manually in the Deskew Editor window.

Once these delays have been determined and set, you can use the Parameter Editor for adjusting the port timing with respect to the system clock (see *"Setting Up Ports and Channels" on page 223*).

NOTE It should be obvious that if you switch the operation mode of an optical-electrical data generator or analyzer from optical to electrical, you have to measure and store the delay values of the electrical cables that are now used for the test.

How to Export / Import Settings or Segments

Settings and segments can be exported as ASCII files. These files can be stored on disk or on diskette.

Export / Import of a Setting

This function serves the following purposes:

- You can create an archive of settings and store it at a secure place.
- If you need one of the settings once again, you can import any of the settings from the archive.
- You can investigate the exported setting with an editor and use it as a template for programming a test.
- If you have downgraded your Agilent 81250 system or frontends have been changed, some of your stored settings may not work any more. Exported settings can be edited and re-imported to fit to the new configuration.

How to Export a Setting

1. Open the *File* menu.
2. Choose *Export Setting* (see *“Export Setting” on page 157*).

How to Import a Setting

1. Open the *File* menu.
2. Choose *Import Setting* (see *“Import Setting” on page 156*).

Contents of a Setting File

A setting file is organized in blocks of firmware commands. The blocks start with a comment line. An example is shown below:

```
// Reset
:MMEM:SETT:NEW

// Create and connect Ports and Terminals
:SGEN:PDAT1:APP "OUTPUT_PORT",2,"Databus"
:SGEN:PDAT1:TERM1:REN "Databus0"
:SGEN:CONN:PDAT1:TERM1:TO (@0102003)
:SGEN:PDAT1:TERM2:REN "Databus1"
:SGEN:CONN:PDAT1:TERM2:TO (@0102004)
:SGEN:PDAT2:APP "INPUT_PORT",1,"Input"
:SGEN:PDAT2:TERM1:REN "Input0"
:SGEN:PPUL1:APP "INPUT_PORT",1,"Clock"
:SGEN:PPUL1:TERM1:REN "Clock0"
:SGEN:CONN:PPUL1:TERM1:TO (@0105001)

// Module type: E4805A
:SGEN:GLOB:TRIG INT10;
:SGEN:GLOB:TRIG:TVOL 0.0E+0;
:SGEN:GLOB:PER 8.333333333333E-9;
:SGEN:GLOB:MUX 4;
:MCL:SOUR ON;
:SGEN:GLOB:DOFF 0.0E+0;
:SGEN:GLOB:ARM IMM;
:SGEN:GLOB:ARM:SENS PLEV;
:SGEN:GLOB:ARM:THR 2.0E-1;
:SGEN:GLOB:ARM:TVOL 0.0E+0;
:TRIG:DEL 0.0E+0;
:TRIG:MUX 1.0E+0;
:TRIG:VOLT 2.5E+0;
:TRIG:VOLT:LOW 0.0E+0;
:TRIG:TVOL 0.0E+0;
:TRIG:IMP 5.0E+1;
:TRIG:MODE CGEN;

// Term type: E4844A
:SGEN:PDAT1:TERM1:MUX 1.0E+0;
:SGEN:PDAT1:TERM1:INP:DEL 4.166667E-9;
:SGEN:PDAT1:TERM1:INP:DEL:CYCL 5.0E-1;
:SGEN:PDAT1:TERM1:INP:DEL:TIME 0.0E+0;
:SGEN:PDAT1:TERM1:INP:THR 0.0E+0;
:SGEN:PDAT1:TERM1:INP:TVOL 0.0E+0;
:SGEN:PDAT1:TERM1:INP:IMP 5.0E+1;
:SGEN:PDAT1:TERM1:INP:SER 0.0E+0;
:SGEN:PDAT1:TERM1:INP ON;
```

Export / Import of Segments

This function serves the following purposes:

- You can create an archive of segments and store it at a secure place.
- You can create segments with an editor and import them.
- Available data patterns can easily be inserted into the test sequence.
- Exported segments can be edited and re-imported.

How to Export Segments

1 Open the *File* menu.

2 Choose *Export Segments* (see *"Export Segments" on page 157*).

How to Import Segments

1 Open the *File* menu.

2 Choose *Import Segments* (see *"Import Segments" on page 156*).

Contents of a Segment File

A segment file can hold several segments. Each segment has a general structure as shown in the example below:

```
:vectorVariablesDefinitions:
{
   :paraPatternVar:


   {
      :name: Input1
      :statePar: {  {A "01"}  }
      :stateSet: A
      :vectorWidth: 1
      :vectors:
         {
      1
      0
      1
      0
      1
      0
      1
      1
      1
      0
      1
      0
         }
         :parameters:
         {
            { _Type (MEMORY) }
         }
      }
   }
```

How to Execute Firmware Commands

The Command Line Editor allows control of an instrument through the command string interface. This window is intended as a test editor to test individual commands for a remote program.

How to Start the Command Line Editor

To start the Command Line Editor:

1 Open the *Go* menu.

2 Choose *Command Line*.

The following window opens:

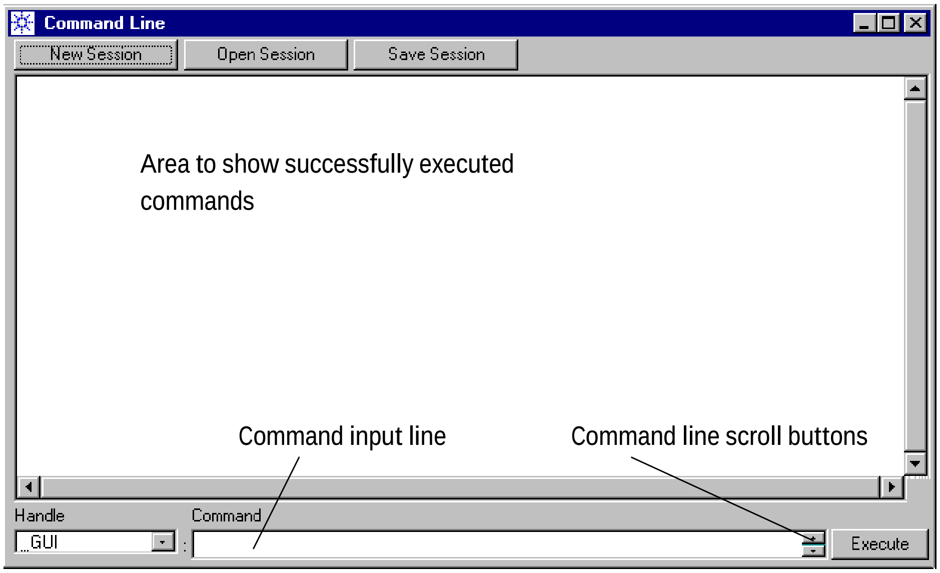


Figure 258 Command Line Editor

The Command Line window is divided into two areas:

- Display area, occupies the upper part of the window.
- Command entry area, occupies the lower part of the window.

Commands can be entered in the command input line. Successful commands and their results are displayed in the display area.

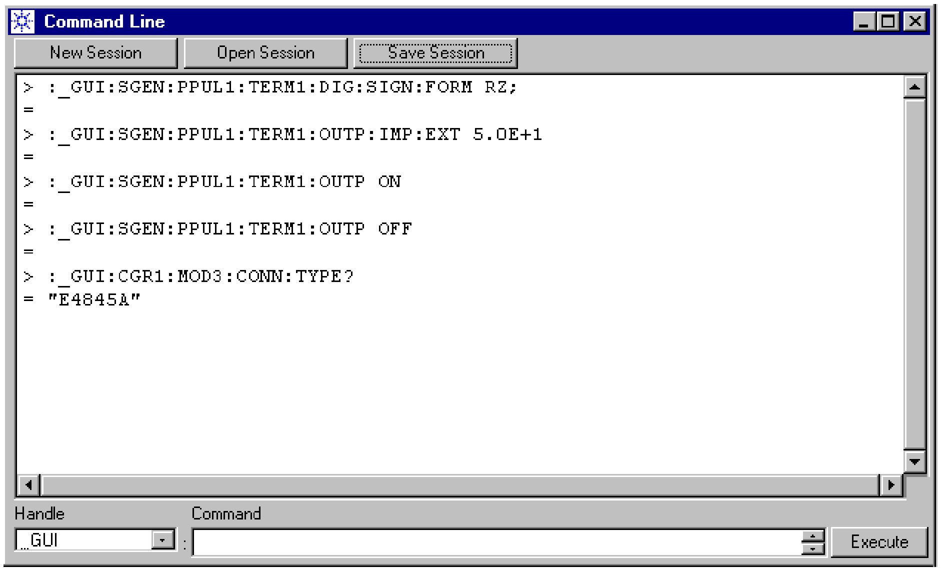
How to Use the Command Line Editor

The commands you enter are transmitted to and executed by the system to which the current user interface is connected (see also *"How to Configure the User Interface" on page 140*).

All commands available for the Agilent 81250 system can be entered. For details see the *Agilent 81250 SCPI Programming Reference*. Note that the first colon is automatically provided and must not be entered.

Once you have entered a command in the command entry area, click the *Execute* button. This downloads the command to the firmware where it is executed.

Commands which have been successfully executed are moved up to the display area.



```
> :_GUI:SGEN:PPUL1:TERM1:DIG:SIGN:FORM RZ;
=
> :_GUI:SGEN:PPUL1:TERM1:OUTP:IMP:EXT 5.0E+1
=
> :_GUI:SGEN:PPUL1:TERM1:OUTP ON
=
> :_GUI:SGEN:PPUL1:TERM1:OUTP OFF
=
> :_GUI:CGR1:MOD3:CONN:TYPE?
= "E4845A"
```

Handle: _GUI

Figure 259 Command Line Editor – Executed Commands

Copy and Paste in the Command Line Editor

Selected text can be cut, copied or pasted from and to the command entry area. The Command Line Editor provides a context menu that supports these operations, and the editor reacts on common keyboard shortcuts (Ctrl+c, Ctrl+x, Ctrl+v). It uses the Windows Clipboard.

You can copy and paste parts of a command or a whole command line. Commands can thus be taken from the display area and entered in the command entry area.

The two arrow buttons at the right-hand side of the command entry line as well as the cursor up/down keys can be used for scrolling through the list of previously successful commands.

TIP You can also paste commands that have been copied to the clipboard from external program files, such as an exported setting file. This works in both directions and enables you also to create program files with any program editor that allows copy and paste.

Buttons of the Command Line Editor

The Command Line Editor provides the following buttons:

- *New Session*: Clears the display area.

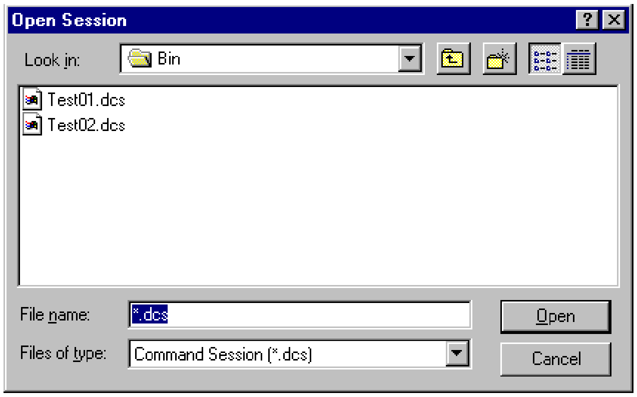
  You can start to create a new sequence of commands to be combined in a new session.

- *Save Session*: Saves the commands shown in the display area in a file on disk.

  The files are saved as command session (*.dcs) files in the c:\hp81200\dsr\bin directory.

- *Open Session*: Used to open a previously saved session and execute it immediately.

Select from the list of available command session files.



Look in: Bin

Test01.dcs
Test02.dcs

File name: *.dcs

Files of type: Command Session (*.dcs)

Figure 260 Command Line Editor – Selection of Saved Sessions

How to Use the System Starter Utilities

It may be necessary for you to operate two or more independent ParBERT 81250 systems simultaneously. This is necessary, for example, if you set up one system to generate data, and another to capture and analyze the data. Or, in production environments, you may want to test several devices parallel to each other and view the results on one screen.

In such cases, you are dealing with several instances of the ParBERT user interface, because each system must be set up individually, and one ParBERT user interface can only control one system.

Using Multiple Systems

The ParBERT system controller can be configured such that the ParBERT User Software starts automatically when the controller is switched on. This is an option of the Agilent 81250 Configuration tool.

When the ParBERT User Software is started, up to ten user interfaces can be started automatically (see *"How to Set the Operating Mode" on page 135*). Simultaneously, complete settings can also be downloaded to the hardware (see *"How to Specify a Start Setting" on page 142*).

If these functions are used, you need only switch the hardware on, and after a while all the systems are ready for running the test. You need only mount the DUT(s).

Now you may want to have a tool that allows you to prepare the run, switch all the connectors off and on, and start/stop the test concurrently on several systems.

For this purpose, the Agilent 81250 Parallel Bit Error Ratio Tester provides two utilities:

- System Starter for 2 Systems

  This tool allows you to start/stop a test on two systems.

- System Starter for n Systems

  This tool allows you to start/stop a test simultaneously on more than two systems. These systems may use different firmware servers.

Both tools can be started from the *Utilities* panel.

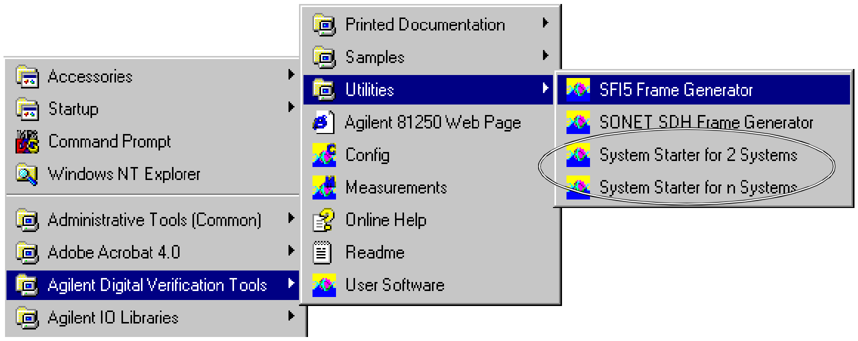


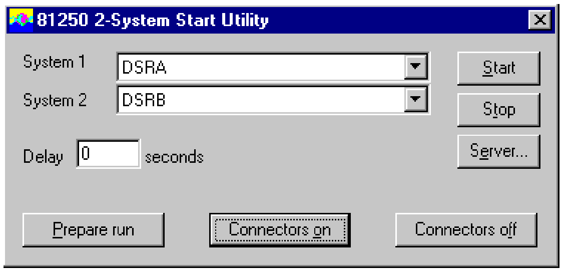
Figure 261 **Agilent Digital Verification Tools Utilities Panel**

NOTE Before starting one of these tools, make sure that a ParBERT firmware server is running. This server can be on your local PC or on any ParBERT controller connected via the LAN.

In addition, ensure that you have set up the systems you wish to control correctly. This is generally done by loading appropriate settings.

Using the System Starter for 2 Systems

This tool is almost self-explaining. It has the following control window:


81250 2-System Start Utility

System 1 DSRA

System 2 DSRB

Delay 0 seconds

Figure 262 **Two Systems Test Start Utility**

Selecting systems 1 and 2 If the ParBERT firmware server is not running on your local PC, or if you wish to connect to a different firmware server, click the Server... button. You can then enter the IP name of the remote ParBERT controller.

The two systems to be operated can be chosen from the drop-down lists. It depends on the firmware server which systems are available.

Performing actions By clicking the appropriate buttons, you can prepare the run, switch all the connectors off and on, start and stop the test.

These actions correspond to the test control buttons of the ParBERT user interface:

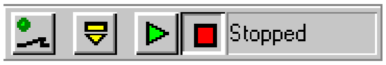
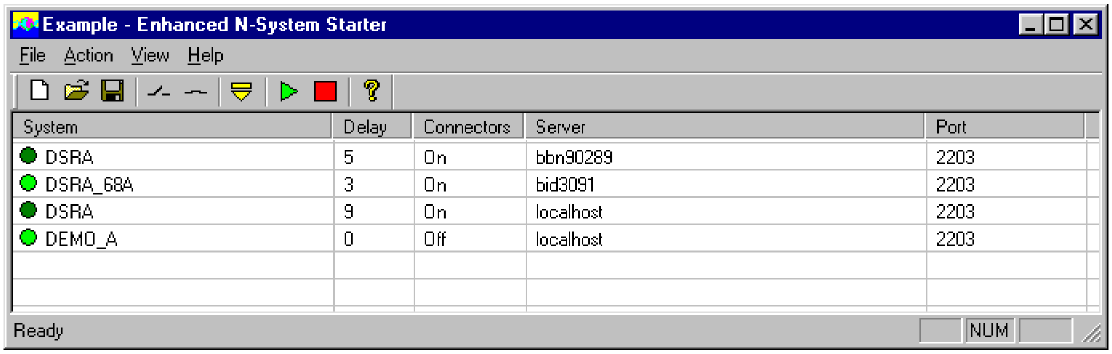


Figure 263 Test Control Buttons

If you specify a *Delay*, the start of the second system will be delayed. This is useful if you, for example, want to start a generator system first, and, after some time, the analyzer system that captures response data.

Using the System Starter for n Systems

The System Starter for n Systems has the following control window:



System	Delay	Connectors	Server	Port
● DSRA	5	On	bbn90289	2203
● DSRA_68A	3	On	bid3091	2203
● DSRA	9	On	localhost	2203
● DEMO_A	0	Off	localhost	2203

Figure 264 N Systems Starter Control Window

The referenced systems can reside on the same controlling host, or on multiple remote hosts, as shown in the figure above.

The status of a system is indicated by the green light left of its name: if the light is on, the system is started, otherwise, it is stopped. When all systems are started at once, their start will be delayed with the delay (in seconds) shown in the *Delay* column. This is useful if you want to start generators first, and, after some time, the analyzers.

Filling the list

To add a system:

1 Open the *Action* menu and choose *Insert new system*

or

double-click in an empty row of the *System* column.

2 Follow the instructions that appear.

Changing the list

To change the list:

1 To move a system in the list, drag its name and drop it over another system. It will be inserted above that system.

2 If you want to put a system at the end of the list, drop it over the free area of the grid (the shape of the cursor changes to an up-arrow).

3 To replace a system of the list by another one, double-click its name.

4 To remove a system from the list, drop it outside the window, or right-click the system name, and select *Delete system*.

Performing actions

The commands of the *Action* menu have right-click equivalents—with an important difference: When you right-click a system's name, it is automatically selected, and the popup menu commands will only affect the selected system (only one system can be selected at a time).

The toolbar buttons are shortcuts for the main menu—so, clicking the Run button will start all systems.

Updating the list

In a remote programming environment, more than one program may connect to the same system and change the system's state without informing the other programs. Therefore, it is possible that the status of a system shown by the System Starter for n Systems does not accurately reflect the real status of the system. If you are not sure that the status of a system is correct, press the F5 key to refresh the current state of all systems in the list.

Saving the list

The system list can be saved on disk for later retrieval (possibly from another computer—the files are not linked to the particular host where the tool was used to create the file).

You may assign any arbitrary file name extension or none at all. The list is stored as plain text. It can be modified by any ASCII editor.

When a list is saved, only the items currently shown on the screen are saved. Systems that were in a loaded list, but are currently generating errors, are not shown, and therefore not saved with a new list.

Therefore, it is recommended that when you make changes to a *loaded* list, you save it under a new name. This makes sure that any systems that were temporarily inaccessible are not deleted from the original list.

Appendix A: How Do I ... ?

This section provides answers to frequently asked questions.

It can well happen that you have a certain problem and just don’t know how to solve it quickly and efficiently with the Agilent 81250 Parallel Bit Error Ratio Tester. It is this kind of problem that is addressed in this chapter.

The chapter covers the following topics:

- *“How Do I Generate a Clock Signal With a Data Module?” on page 390*
- *“How Do I Use Events?” on page 392*
- *“How Can I Change all Traces of a Port to Don’t Care?” on page 397*
- *“How Do I Set Up a Multiplexer BER Test?” on page 400*
- *“How Do I Use Automatic Sampling Point Adjustment?” on page 403*
- *“How Do I Use the AUX OUT of Analyzer Frontends?” on page 412*
- *“How Do I Calibrate a 43G Pattern Generator?” on page 413*

How Do I Generate a Clock Signal With a Data Module?

The data generator/analyzer module must be equipped with one of the generator frontends:

- E4838A, 675 MHz, differential output, low voltage amplitude/offset and variable slopes generator
- E4843A, 675 MHz, NRZ/RZ, differential output frontend
- E4864A, 1.3 Gbit/s, differential NRZ output frontend
- E4862A, 2.7 Gbit/s, differential NRZ output frontend
- E4868A, 3.35 Gbit/s, NRZ/RZ, differential output frontend

Alternatively, you can also use the E4866A 10.8 Gbit/s data generator module.

Using the Pulse Port

The most comfortable way to apply a clock signal to the DUT is:

1. Use the Connection Editor and create a **pulse port**.
2. Connect the pulse terminal to the generator connector you wish to use.
3. Use the Parameter Editor to adjust the channel properties, such as frequency, pulse width, delay, voltage levels, expected load, and so on. Keep the default RZ (Return to Zero) data format. Switch the generator output on.

The E4862A or E4864A generator frontends do not support the RZ data format. If such a generator is connected to a pulse port, the software switches that frontend to “clock mode”. This means that this frontend will automatically generate a pulse with the specified frequency, amplitude, and 50 % duty cycle.

Using a Data Input Port

On the other hand, you can also apply a clock signal to one or several terminals of a **data port**. This is useful if you need to generate a burst of clock pulses.

1 Create the sequence and insert the segments.

2 Use the Segment Editor (easily called from the Sequence Editor or Data/Sequence Editor) to set the data bits of the desired trace(s) to "1".

3 Use the Parameter Editor for the respective channel(s) and select "RZ" (Return to Zero) as the format.

In RZ format every logical 1 creates an electrical pulse, as illustrated in the figure below:

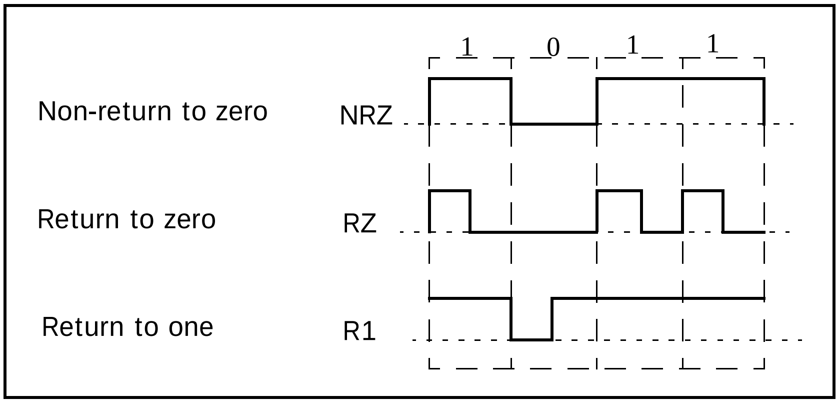


Figure 265 Signal Formats

NOTE The RZ data format is not supported by all generator frontends. For frontends that do not support the RZ format, you would have to create a trace like 1 0 1 0 1 0 etc. If this is done, the resulting pulse has only half the port frequency.

4 If necessary, adjust the channel properties, such as pulse width, delay, voltage levels, termination resistors, and so on. Switch the generator output on.

NOTE This procedure covers one block of the overall test sequence. If the clock signal is to be generated during execution of another block that references different data segments, the steps 2 and 3 have to be repeated for that block.

How Do I Use Events?

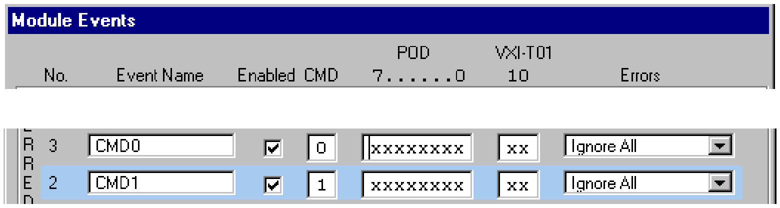
The built-in features for detecting events and reacting upon events provide many capabilities (see *"Event Handling Principles" on page 104*). The characteristics and procedure are explained in *"How to Specify Events and Reactions Upon Events" on page 298*.

This section shows and explains some examples.

How Do I Select Between Two Different Tests?

With this setup you can switch very fast between two different test sequences using the command control feature. With the trigger pod and external hardware even more blocks could be selected.

1 Define the deferred events CMD0 and CMD1.

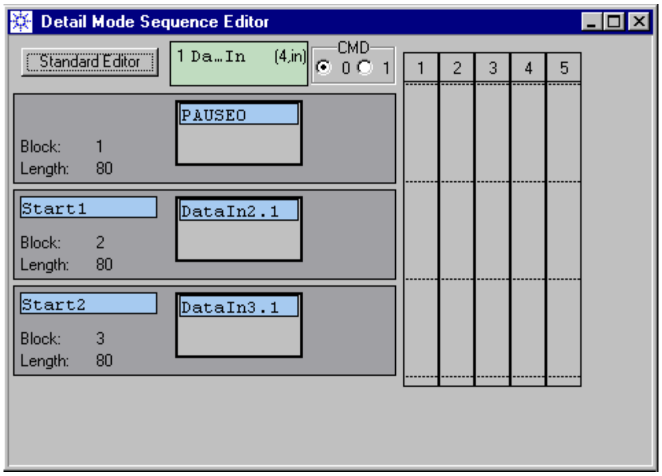


Module Events

No.	Event Name	Enabled	CMD	POD 7.......0	VXI-T01 10	Errors
3	CMD0	☑	0	xxxxxxxx	xx	Ignore All
2	CMD1	☑	1	xxxxxxxx	xx	Ignore All

Figure 266 Definition of 2 Command Events

2 Create the sequence and fill in the branch tables of the blocks.



PAUSE0
Block: 1
Length: 80

Start1
DataIn2.1
Block: 2
Length: 80

Start2
DataIn3.1
Block: 3
Length: 80

If	Go to	Trig.	VXI-T01
CMD0	Start1	0	00
CMD1	Start2	0	00

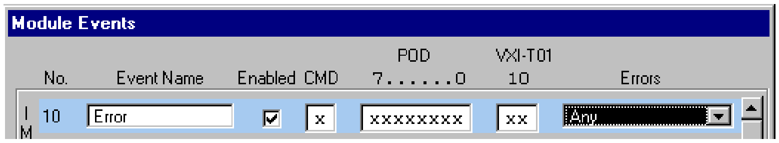
CMD0	END	0	00
CMD1	END	0	00

Figure 267 Sequence and Branch Tables for Selecting One of Two Tests

How Do I Set a Trigger on Error?

We simply want to issue a trigger signal at the TRIGGER OUTPUT of the master clock module whenever an error is detected somewhere in the system. Note that we use an immediate event here.

1 Define the immediate error event.

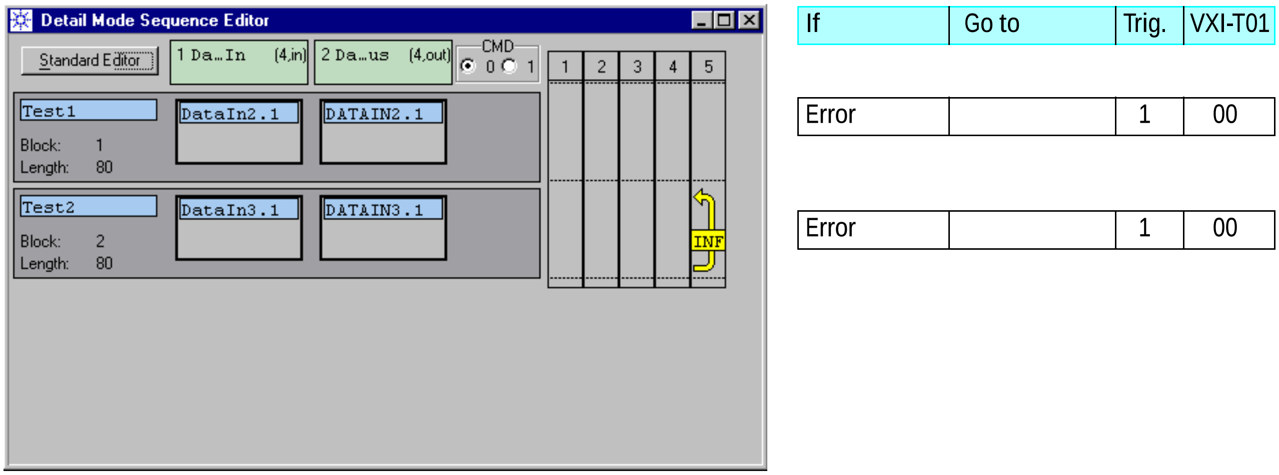


Module Events

No.	Event Name	Enabled	CMD	POD 7......0	VXI-T01 10	Errors
I M 10	Error	☑	x	xxxxxxxx	xx	Any

Figure 268 Definition of an Immediate Error Event

2 Create the sequence and fill in the branch tables.



Block: 1
Length: 80

Block: 2
Length: 80

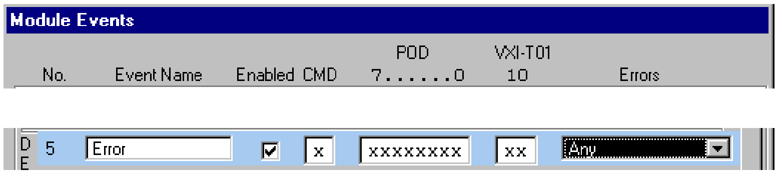
If	Go to	Trig.	VXI-T01
Error		1	00
Error		1	00

Figure 269 Sequence and Branch Tables for Triggering on Error

How Do I Allow the DUT to Stabilize?

In this example, we wait until we have no error condition for a certain time (determined by the block length).

1 Define a deferred error event.

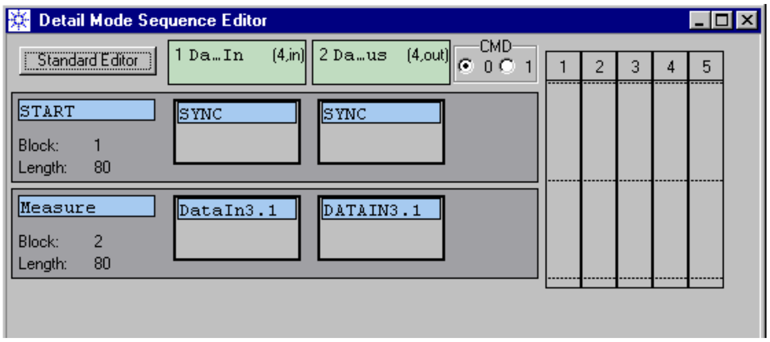


Module Events

No.	Event Name	Enabled	CMD	POD 7......0	VXI-T01 10	Errors
D E 5	Error	☑	x	xxxxxxxx	xx	Any

Figure 270 Definition of a Deferred Error Event

2 Create the sequence and fill in the branch table of the START block.



Detail Mode Sequence Editor

Standard Editor | 1 Da…In (4,in) | 2 Da…us (4,out) | CMD ● 0 ○ 1 | 1 2 3 4 5

START
Block: 1
Length: 80
SYNC | SYNC

Measure
Block: 2
Length: 80
DataIn3.1 | DATAIN3.1

If	Go to	Trig.	VXI-T01
Error	START	0	00

Figure 271 Sequence and Branch Table for a Test With Warm-Up

The START block will be repeated until it causes no error.

How Can I Return Pass / Fail Information to another Test System?

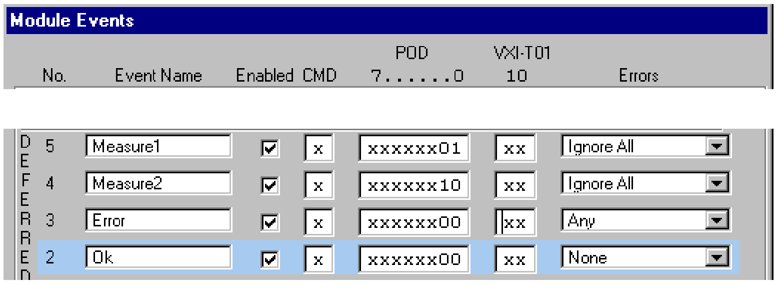
In this bolt-on example, our instrument is integrated into a large IC test system. We shall run a measurement that is selected via the trigger input pod.

The result is a pass/fail signal that is returned to and examined by the large IC test system.

A generator frontend will be used to generate the pass/fail signal. The output of this generator (low or high) is defined by two memory segments. One of these segments contains only zeros, the other only ones.

In order to use the generator, we have specified a one-terminal data input port. This port does actually not belong to the DUT, because the generator's output is physically connected to a sense-pin of the IC test system.

1 Define four events.



Module Events

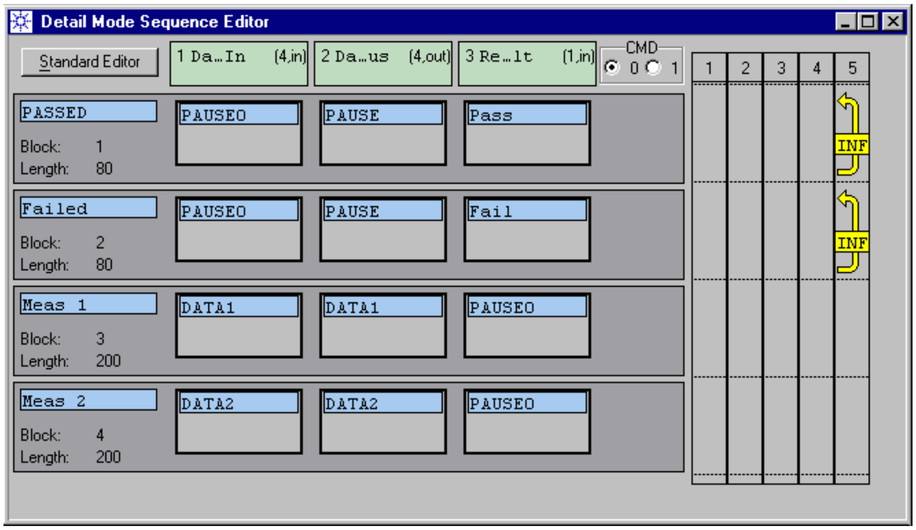
DEFERRED

No.	Event Name	Enabled	CMD	POD 7......0	VXI-T01 10	Errors
5	Measure1	☑	x	xxxxxx01	xx	Ignore All
4	Measure2	☑	x	xxxxxx10	xx	Ignore All
3	Error	☑	x	xxxxxx00	xx	Any
2	Ok	☑	x	xxxxxx00	xx	None

Figure 272 Definition of Pod and Error Events

2 Create the sequence and fill in the branch tables.

Remember that we use a one-terminal data input (= generator output) port and two segments to generate and return the pass/fail signal. The segments are called PASS and FAIL.



	1 Da…In (4,in)	2 Da…us (4,out)	3 Re…lt (1,in)
PASSED Block: 1 Length: 80	PAUSE0	PAUSE	Pass
Failed Block: 2 Length: 80	PAUSE0	PAUSE	Fail
Meas 1 Block: 3 Length: 200	DATA1	DATA1	PAUSE0
Meas 2 Block: 4 Length: 200	DATA2	DATA2	PAUSE0

If	Go to
Measure1	Meas_1
Measure2	Meas_2

Measure1	Meas_1
Measure2	Meas_2

Error	Failed
Ok	PASSED

Error	Failed
Ok	PASSED

Figure 273 Sequence and Branch Tables

Remarks

The MEAS_1 and MEAS_2 blocks need to be longer than the actual measurement so that all analyzer pipelines are toggled through and the events are completely processed.

The event definitions shown above require that the trigger pod inputs are driven with positive pulses to select the measurement blocks. During the measurement, the pod input lines must be zero, otherwise no errors are detected. This should not be a problem because due to the internal pipelining, it takes a couple of sequencer clock periods until the first error can be detected (see also *"What You Need to Consider Before Using Events" on page 300*).

The events that cause the jump to the different measurement blocks can be deferred or immediate. The Ok and Error events, however, must be deferred, and the Error event must have a higher priority than the Ok event.

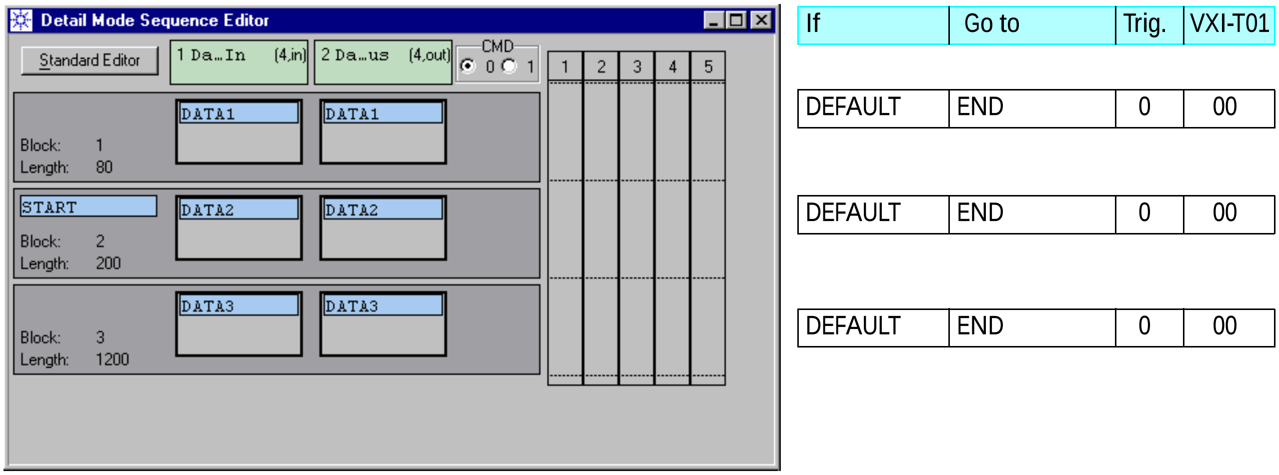
If you want to save events, the Ok event could be replaced by the DEFAULT event.

How Can I Execute Different Tests Within One Sequence?

Starting at a certain block label is fairly simple—the CMD 0/1 command, the VXI trigger lines, or the trigger pod can be used to switch between several start blocks.

The following example shows how the sequence can be terminated after executing a certain block.

We use the DEFAULT event which does not have to be defined. It occurs automatically at the end of the block.



DATA1 DATA1
Block: 1
Length: 80

START
DATA2 DATA2
Block: 2
Length: 200

DATA3 DATA3
Block: 3
Length: 1200

If	Go to	Trig.	VXI-T01
DEFAULT	END	0	00
DEFAULT	END	0	00
DEFAULT	END	0	00

Figure 274 **Sequence and Branch Tables**

In this example, only block 2 is executed. By moving the START label, any other block can be executed.

How Can I Change all Traces of a Port to Don’t Care?

The simplest way to ignore incoming data is to use a Pause segment.

But for data output (= analyzer input) ports, don’t care setting is also available. That makes it possible to exclude portions of the incoming data from analysis.

The following procedure can be used if the measurement mode is set to Compare and Capture or to Compare and Acquire around Error.

Use the Detail Mode Sequence Editor or the Data/Sequence Editor.

To replace the present segment by a Don’t Care pseudo segment:

1 Open the segment’s context menu.

2 Click *Don’t Care*.

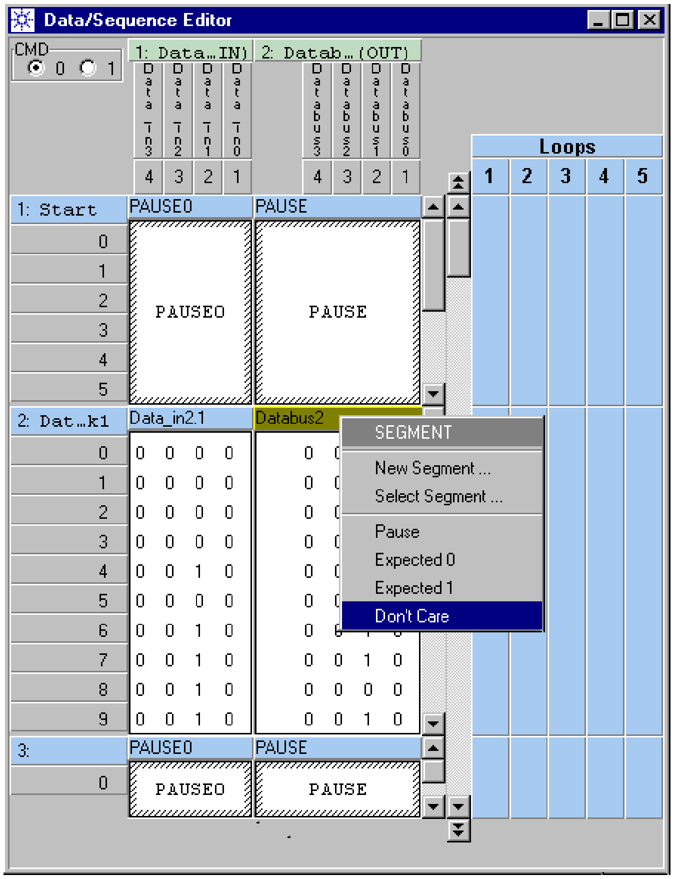


Figure 275 Segment Context Menu

This changes the whole segment.

To change individual vectors or traces:

1 Right-click the segment’s edit area.

This opens the Segment Editor context menu.

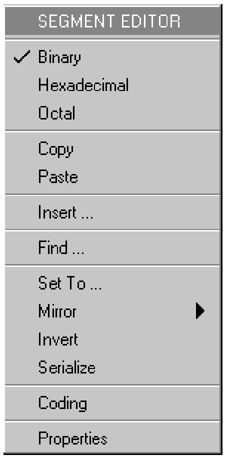


Figure 276 **Segment Editor Context Menu**

2 Click *Coding* or *Properties* and ensure that the state coding is set to 0x1.

Only 0x1-segments can have don’t care settings. Such segments use two memory bits for each data bit.

3 Highlight the traces or vectors you wish to change.

4 Open the Segment Editor context menu once more and click *Set To*.

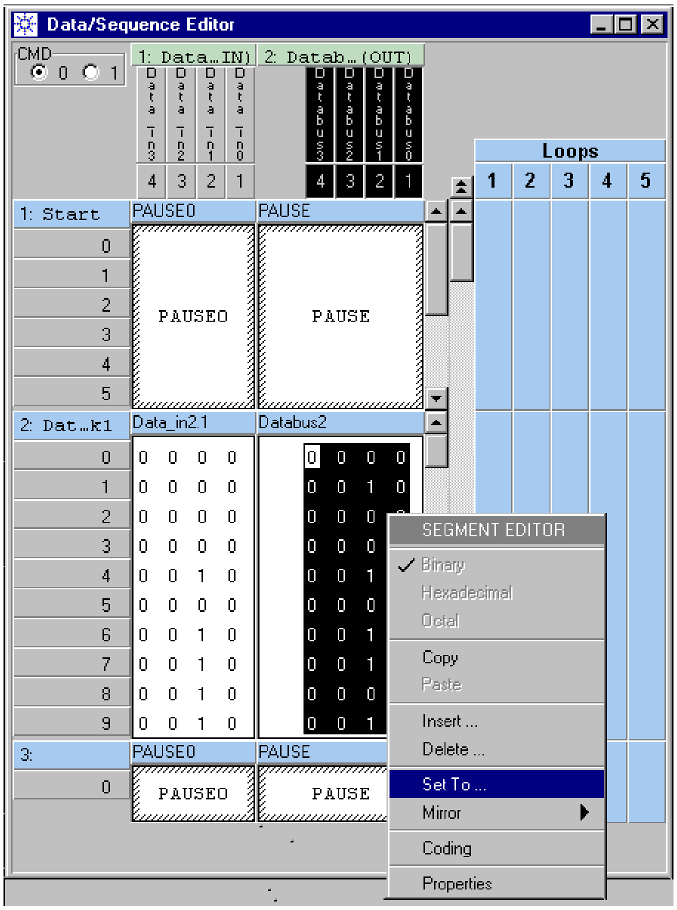


Figure 277 **Segment Editor Context Menu after Highlighting Data**

5 Choose x from the pull down menu.

This changes all selected items to Don’t Care.

How Do I Set Up a Multiplexer BER Test?

The bit error rate (BER) of a multiplexer can be measured by sourcing a PRWS (pseudo random word stream) segment to the DUT and comparing the serial output with a PRBS (pseudo random bit stream) segment of the same order.

The connections could be made as shown below:

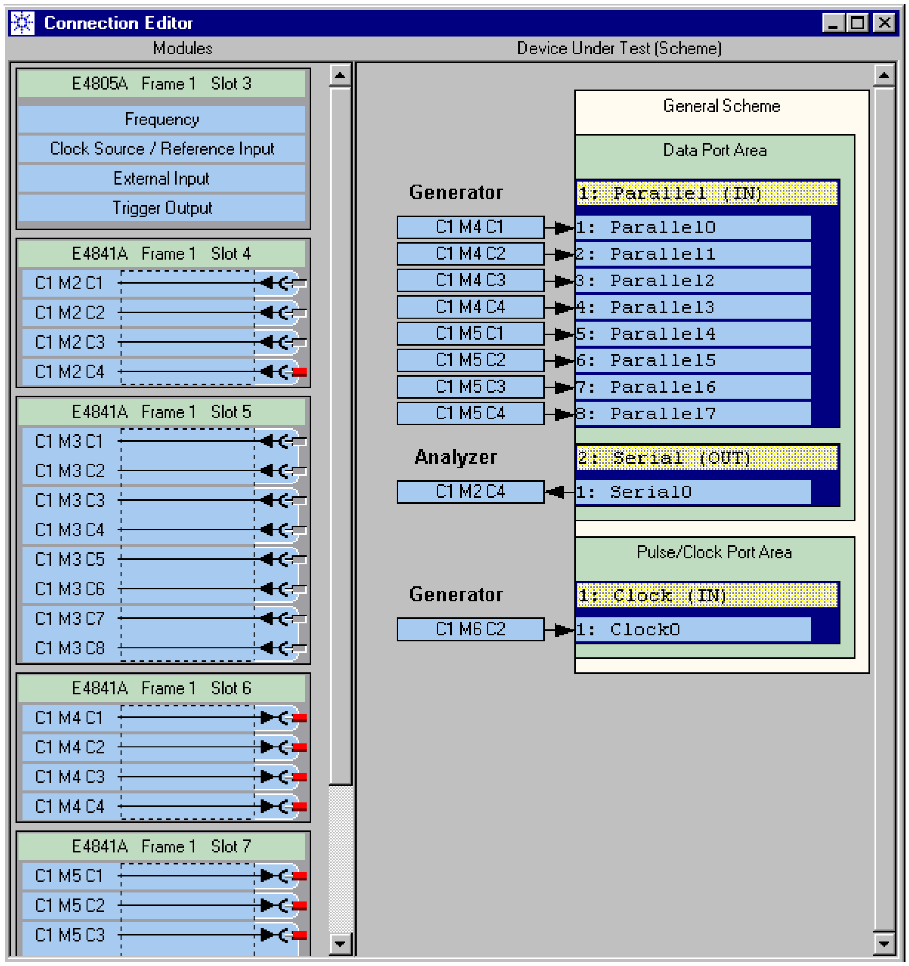
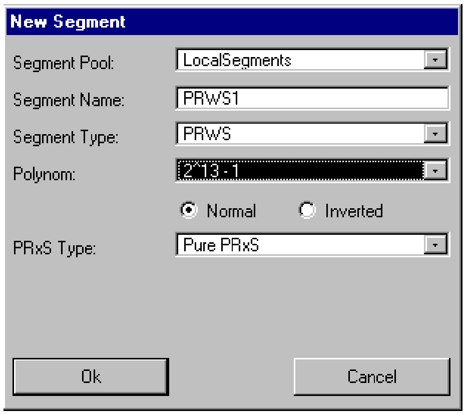


Figure 278 Connections for a Multiplexer Test

1 Create a PRWS segment to be used for the DUT input port.



New Segment

Segment Pool: LocalSegments

Segment Name: PRWS1

Segment Type: PRWS

Polynom: 2^13-1

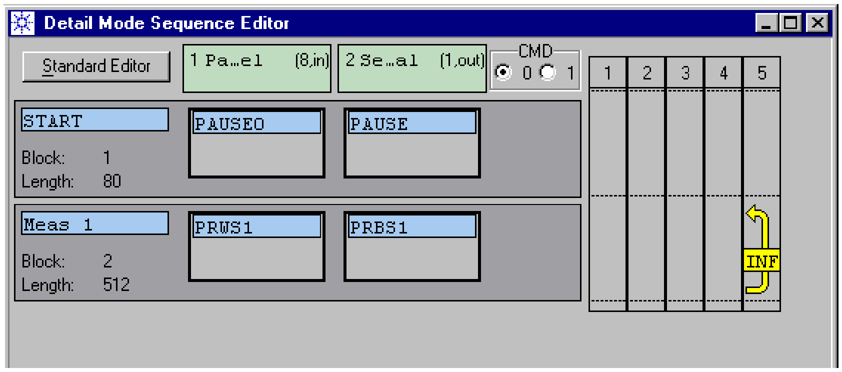
● Normal ○ Inverted

PRxS Type: Pure PRxS

Figure 279 Segment Editor Window

2 Create a PRBS segment of the same order (the same polynomial) to be used for the DUT output port.

3 Use the Sequence Editor and insert the segments into the sequence.



START

Block: 1
Length: 80

PAUSE0

PAUSE

Meas 1

Block: 2
Length: 512

PRWS1

PRBS1

Figure 280 MUX Test Sequence

The correspondence between the generated PRWS and the expected PRBS is as follows:

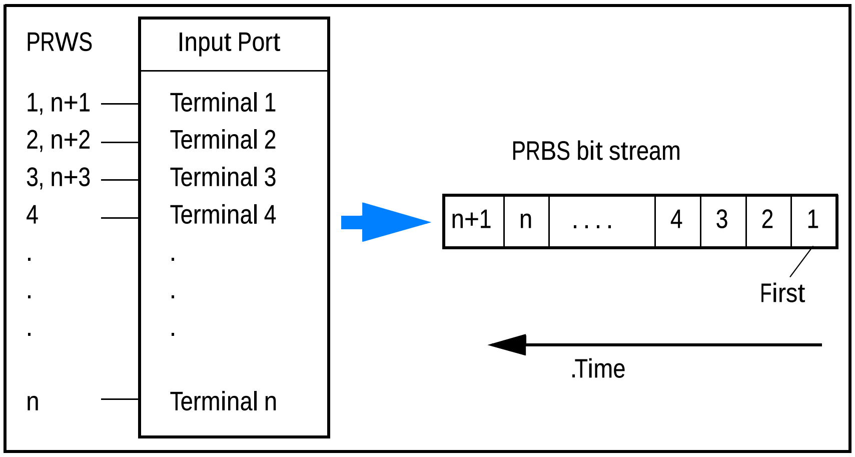


Figure 281 Correspondence Between Random Words and Random Bit Stream

The bits of the PRWS are assigned to the generator channels from top to bottom, as displayed by the Connection Editor.

The PRBS contains the same logical bit sequence, just one-directional.

The same principle—only inverted—applies if you are testing a demultiplexer.

NOTE Note that the assignment of bits to generator channels is different, if you apply the PRWS to channels that have been digitally added.

In this case, the system assumes that the word length *n* is equal to the total number of channels involved. It assigns the first *m* bits to the connected channels and the remaining *n* – *m* bits to the added, unconnected channels.

Example If you had an input port with five terminals and the first two terminals were connected to two added channels, then terminal 1 would receive the XOR of bits 1 and 6 and terminal 2 the XOR of bits 2 and 7, as illustrated below:

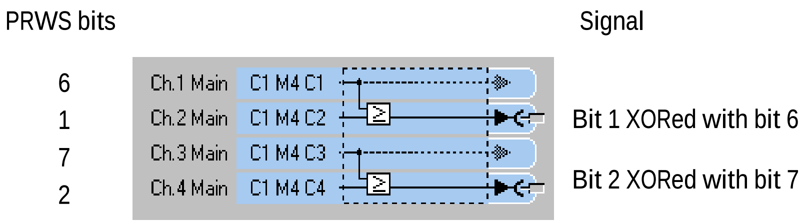


Figure 282 Bit Assignment to Digitally Added Channels

How Do I Use Automatic Sampling Point Adjustment?

The functions Automatic Bit Synchronization and Automatic Delay Alignment can be used to adjust the sampling point of the analyzers of an Agilent 81250 system that sources data to and simultaneously captures data from the DUT (see also *"Principles of Analyzer Sampling Point Adjustment" on page 87*).

However, these functions can also be used for synchronizing two separate systems. Separate systems with individual clocks are required if, for example, multiplexers/demultiplexers are to be tested that have a mux-factor other than 2^n.

This section gives some examples of setups and procedures.

How Can I Synchronize a MUX Test With Two Systems?

The functions for automatic analyzer sampling point adjustment require that certain conditions are met.

Requirements for MUX tests with two systems:

Automatic Delay Alignment	Delay window is known and can be specified.
Automatic Bit Synchronization	PRxS data may be sent and expected. If memory-type data is used, then the first 48 bits of every segment trace must form a unique word.

Using Automatic Delay Alignment

Test Setup

The analyzer system is in the external clock mode and starts with an external trigger provided by the generator system:

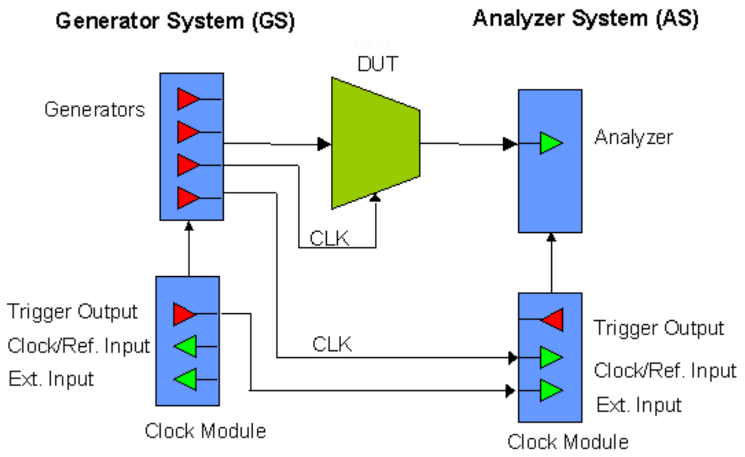


Figure 283 MUX Test Setup With Automatic Delay Alignment

Characteristics

- GS to AS frequency ratio flexible (using the clock module's clock input multiplier/divider).
- All kinds of patterns can be used.
- Delay window must be known (±10ns E4861A, ±50ns E4832A).
- Analyzer start delay can be set to roughly place delay window.
- Returns the absolute delay in the Parameter Editor window.

Procedure

1. Connect the DUT in the Connection window.
2. Set the appropriate levels.
3. Set up the data to be sent and expected.

   The data of the first block should not be PRBS. If PRBS is required, set the block length to $(2^n-1) \times$ *Segment Resolution* to ensure that the GS and AS PRBS phases match.

4 Enable the Auto Delay Alignment.

The sequence flow is illustrated below.

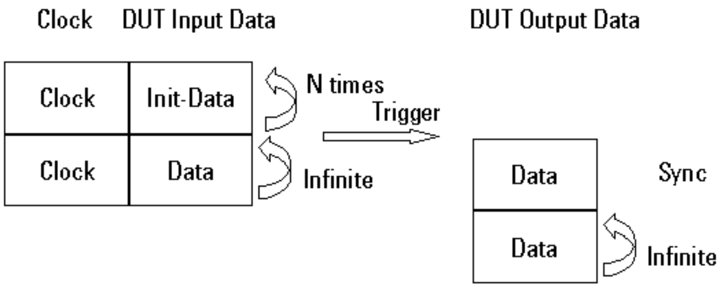


Figure 284 Sequence Flow for Automatic Delay Alignment

5 Set the repetition counter of the start block to a number (N) that suffices to allow the PLL of the analyzer to lock on the generator clock (see ParBERT specifications).

6 Set the analyzer system to external start. Start the analyzer system. It now waits for the trigger.

7 Start the generator system.

The generator sends data and clock to the DUT and an additional clock to the analyzer. After N repetitions of the start block, the generator issues the start trigger and begins an infinite loop.

The N repetitions give the analyzer's PLL time to lock to the external reference clock. When the trigger arrives, the analyzer becomes active. It waits for the specified start delay and then begins sampling the incoming data. The Automatic Delay Alignment assures that the incoming data is sampled at the optimum point of time.

Now the analyzer enters the second block and performs the measurement.

Using Automatic Bit Synchronization With PRBS Data

Test Setup

A multiplexer test with two systems can be set up as shown below:

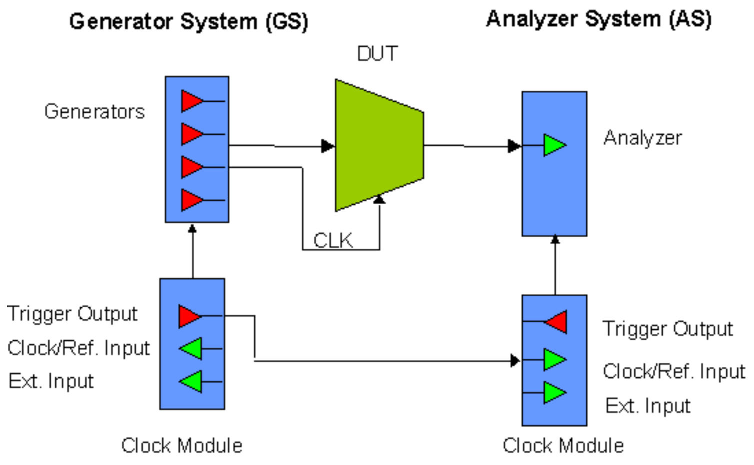


Figure 285 MUX Test Setup With Automatic Bit Synchronization

One system is used to source clock and data to the multiplexer and an additional clock to the analyzer. The other system is used for data analysis.

Characteristics

- GS to AS frequency ratio is flexible (using the clock module’s clock input multiplier/divider).
- No delay window must be known.
- No special initialization block is required.
- No special block length restrictions.
- No absolute delay known afterwards (uncertainty of n periods).
- The timing between analyzers can be compared for calculating skews.
- Usually faster than Automatic Delay Alignment

Procedure

1. Connect the DUT in the Connection window.
2. Set the appropriate levels.
3. Use the Standard Mode Sequence Editor to set up the data to be sent and expected.
4. Start the generator.
5. Wait until the PLL of the DUT has stabilized.
6. Start the analyzer system.

Using Automatic Bit Synchronization With Memory Data

Test Setup This is the same setup as in the previous example.

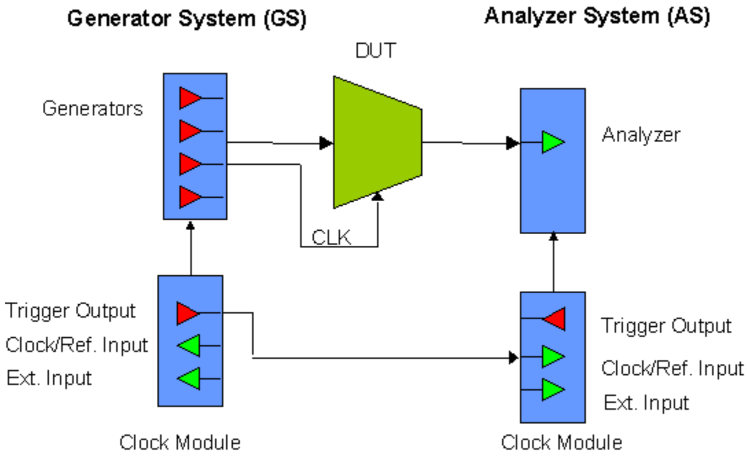


Figure 286 **MUX Test Setup With Automatic Bit Synchronization**

Characteristics Auto Bit Sync with memory data requires some attention:

- Non-pure PRBS are memory segments with a block length of $(2^n\text{-}1) \times$ *segment resolution*.
- The first 48 bits of each segment trace must be unique.
- No absolute timing information is available after synchronization.
- Can only be performed on an independent analyzer system.
- Requires a minimum block length 32 × *segment resolution* × *frequency multiplier*.
- All detect words of all channels must be found within ±5 × *segment resolution*.

Procedure

1 Connect the DUT in the Connection Editor.

2 Set the appropriate levels.

3 Use the Detail Mode Sequence Editor to set up the data to be sent and expected. Take care of the block length.

4 Use the Trigger Pod, if you wish to react on external events.

5 Start the generator.

6 Wait until the PLL of the DUT has stabilized.

7 Start the analyzer system.

How Can I Synchronize a DEMUX Test With Two Systems?

The functions for automatic analyzer sampling point adjustment require that certain conditions are met.

Requirements for DEMUX tests with two systems:

Automatic Delay Alignment	Delay window is known and can be specified. Word phase delay is known.
Automatic Bit Synchronization	PRxS data may be sent and expected. If this is not pure PRxS, the word phase delay must be known. If memory-type data is used, then: - the first 48 bits of every trace must be unmistakable - the word phase delay must be known.

Using Automatic Delay Alignment

Test Setup

The analyzer system is in the external clock mode and starts with an external trigger:

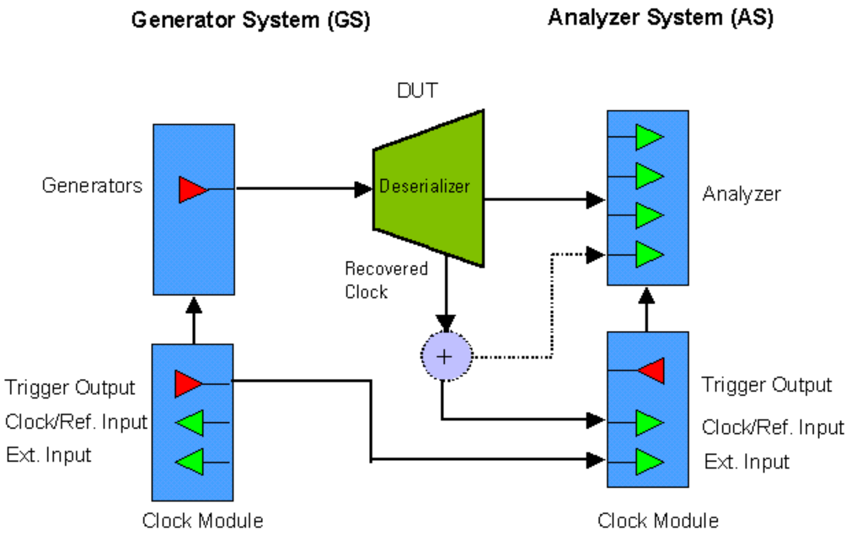


Figure 287 DEMUX Test Setup With Automatic Delay Alignment

Characteristics

- GS to AS frequency ratio is flexible (using the clock module’s clock input multiplier/divider).
- All kinds of patterns can be used.

- Delay window must be known (±10ns E4861A, ±50ns E4832A)
- Analyzer start delay can be set to roughly place delay window.
- Returns the absolute delay in the Parameter Editor window.

Procedure

1 Connect the DUT in the Connection window.

2 Set the appropriate levels.

3 Set up the data to be sent and expected.

The data of the first block should not be PRBS. If PRBS is required, set the block length to (2^n-1) × Segment Resolution to ensure GS and AS PRBS phases match.

4 Enable the Auto Delay Alignment.

The sequence flow is illustrated below.

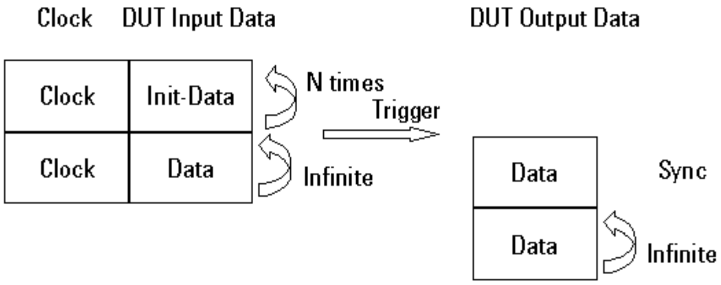


Figure 288 Sequence Flow for Automatic Delay Alignment

5 Set the repetition counter of the start block to a number (N) that suffices to allow the PLL of the analyzer to lock on the generator clock (see ParBERT specifications).

6 Set the analyzer system to external start. Start the analyzer system. It now waits for the trigger.

7 Start the generator system.

The generator sends data and clock to the DUT and an additional clock to the analyzer.

The synchronization block is repeated until the DUT’s and the analyzer’s PLLs have stabilized.

Using Automatic Bit Synchronization With PRBS Data

Test Setup A demultiplexer test with two systems can be set up as shown below:

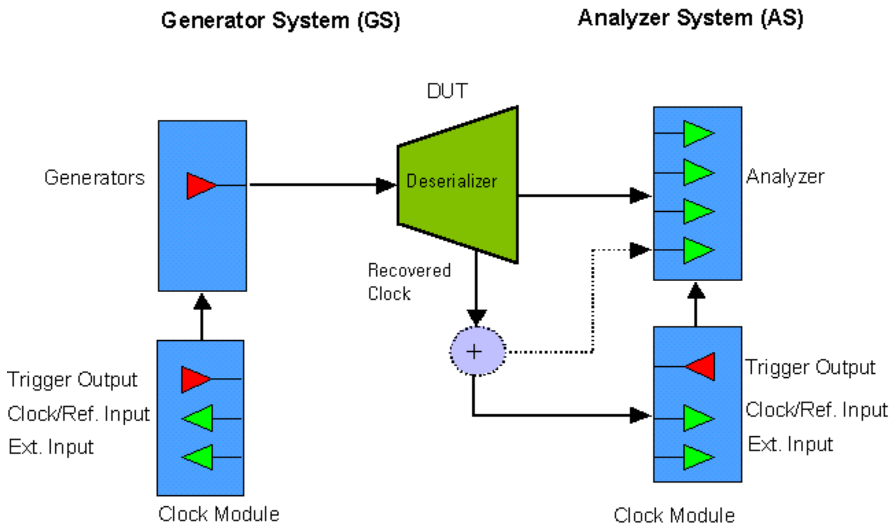


Figure 289 **DEMUX Test Setup With Automatic Bit Synchronization**

One system is used for data generation. The other system is used for data analysis.

The analyzer runs in “external clock” mode. The demultiplexer provides this clock. The frequency of the external clock has to be programmed or measured by the analyzer system to enable the system to establish the correct sample point delays.

If you want to obtain the timing of this measurement relative to the clock of the DUT, then you have to feed the clock recovered by the demultiplexer via a power splitter to the clock module and to an additional analyzer channel.

Characteristics

- No delay window must be known.
- No special initialization block is required.
- No special block length restrictions.
- No absolute delay known afterwards (uncertainty of n periods).
- The timing between analyzers can be compared for calculating skews.
- Usually faster than Automatic Delay Alignment

Procedure

1 Connect the DUT in the Connection window.

2 Set the appropriate levels.

3 Use the Standard Mode Sequence Editor to set up the data to be sent and expected.

4 Start the generator.

5 Wait until the PLL of the DUT has stabilized.

6 Start the analyzer system.

Using Automatic Bit Synchronization With Memory Data

Test Setup

This is the same setup as in the previous example.

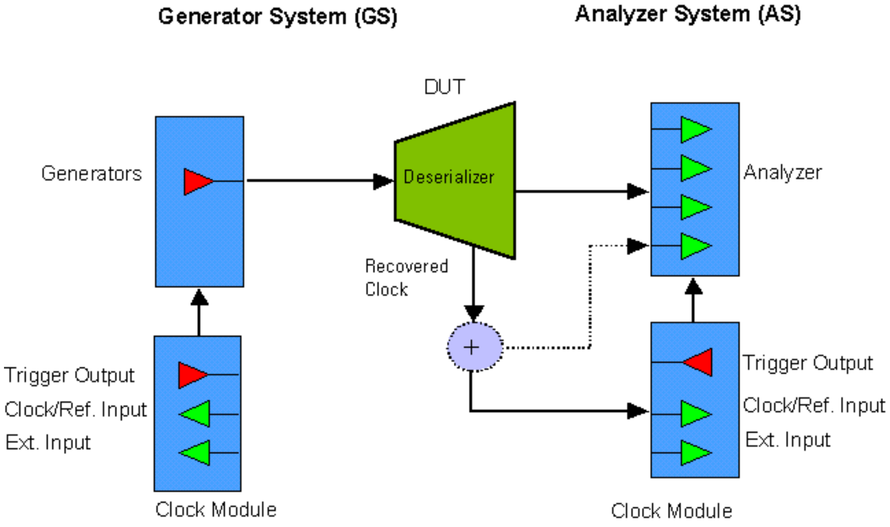


Figure 290 DEMUX Test Setup With Automatic Bit Synchronization

Characteristics

Auto Bit Sync with memory data requires some attention:

- Non-pure PRBS are memory segments with a block length of (2^n-1) × *segment resolution*.
- The first 48 bits of each segment trace must be unique.
- No absolute timing information is available after synchronization.
- Can only be performed on an independent analyzer system.
- Requires a minimum block length 32 × *segment resolution* × *frequency multiplier*.
- All detect words of all channels must be found within ±5 × *segment resolution*.

Procedure

1. Connect the DUT in the Connection Editor.
2. Set the appropriate levels.
3. Use the Detail Mode Sequence Editor to set up the data to be sent and expected. Take care of the block length.
4. Use the Trigger Pod, if you wish to react on external events.
5. Start the generator.
6. Wait until the PLL of the DUT has stabilized.
7. Start the analyzer system.

How Do I Use the AUX OUT of Analyzer Frontends?

The analyzer frontends E4863A/B and E4865A have an AUX OUT connector.

Setup Example

This connector allows you to route a recovered clock signal to the clock module.

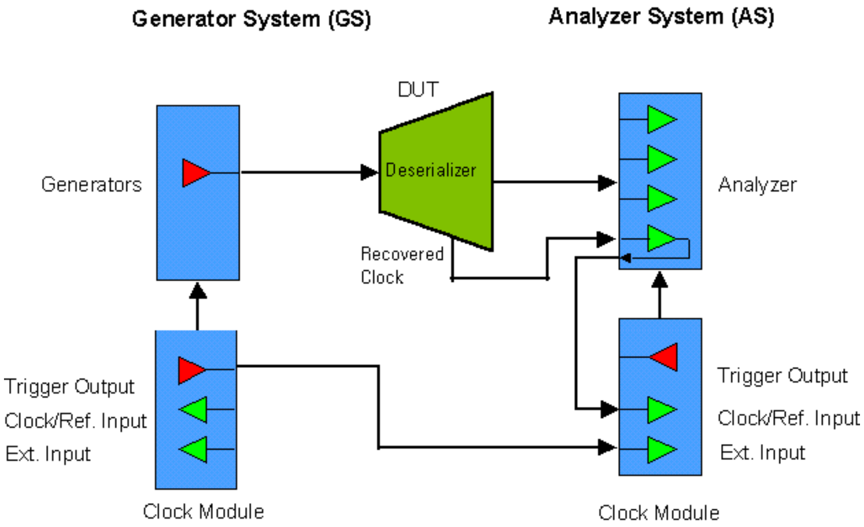


Figure 291 DEMUX Test Setup With Recovered Clock

The AUX OUT connector is connected to the output of the analyzer’s comparators and delivers a unipolar signal.

In the example above, the analyzer system is started by a trigger issued from the generator system. It then uses the recovered clock of the DUT to set its own capturing frequency.

The AUX OUT has an internal impedance of 50 Ω and has to be terminated accordingly.

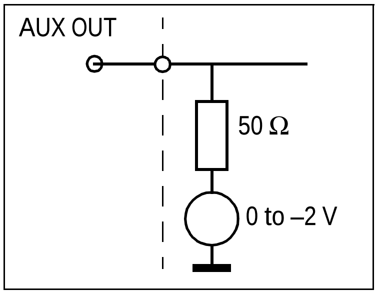


Figure 292 **AUX OUT Termination**

A termination voltage between 0 V and –2 V may be used.

How Do I Calibrate a 43G Pattern Generator?

When you receive a new ParBERT 43G pattern generator, this is factory-calibrated and ready for use.

Recalibration is required in the following cases:

- The E4808A clock module has been replaced.
- The E4868A MUX module has been replaced.
- The cables between the data generator frontends and the MUX module have been replaced by cables with different characteristics.
- An external source clock is connected to the MUX module and the cable between the MUX module's "Sys Clk Output" and the clock module's CLOCK/REF INPUT has been replaced by a cable with different characteristics.

Delay calibration of the ParBERT 43G pattern generator ensures that the sixteen generated signals arrive at the inputs of the MUX module synchronized to the module's setup and hold time for all data rates from 38 Gbit/s up to 43.2 Gbit/s.

The calibration adds a common delay to all sixteen generators, so that the signals seen by the MUX module are sampled at the optimum point of time.

This section explains how to calibrate the ParBERT 43G pattern generator.

NOTE As the system keeps two mutually independent MUX module calibration values—one if an external source clock is used, and one if the clock module generates the clock—the calibration has to be performed twice, if:

- The MUX module was exchanged
- Cables between the data generator frontends and the MUX module were replaced by cables with different characteristics

Prerequisites

The following is needed:

- ParBERT 43G calibration setting (delivered with the software)
- ParBERT 43G calibration data segment (delivered with the software)
- External precision clock generator (for example Agilent E8244A)
- Oscilloscope

Setup

Proceed as follows:

1. Import the calibration segment and setting.

   To locate the segment and the setting from the Windows desktop, click: *START – Programs – Agilent Digital Verification Tools – Samples – Settings – E4860A_Deskew Calibration*. The file names are `Calibration_SEG.txt` and `Calibration_SETT.txt`.

   See also *“How to Export/Import Settings or Segments” on page 378*.

2. Load the calibration setting.

3. Connect the external clock instrument to the “10.8/21.6 GHz Ext. Clock Input” of the MUX module. Set the instrument to 10.8 GHz.

4 Connect the "Sys Clk Output" of the MUX module to the CLOCK/REF INPUT of the clock module.

The Sys Clk Output connector delivers a pulse of 2.7 GHz (1/4th of the 10.8 GHz external clock).

The following figure shows the connections:

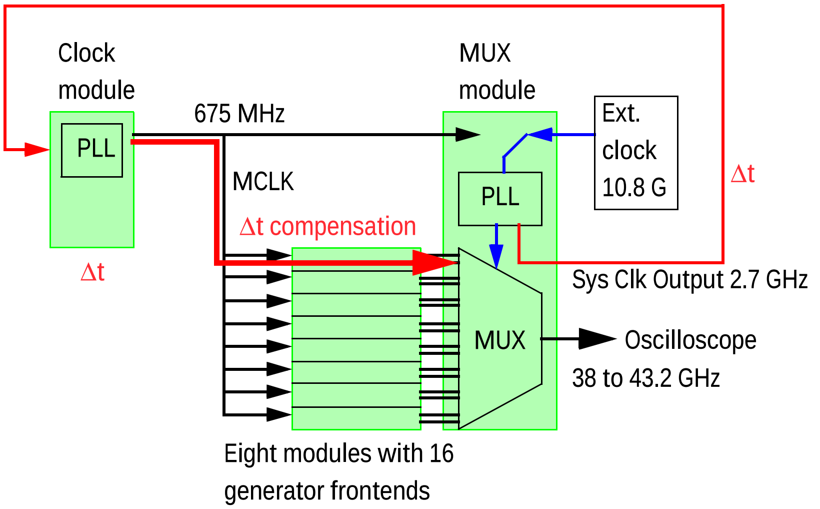


Figure 293 ParBERT 43G Pattern Generator Setup

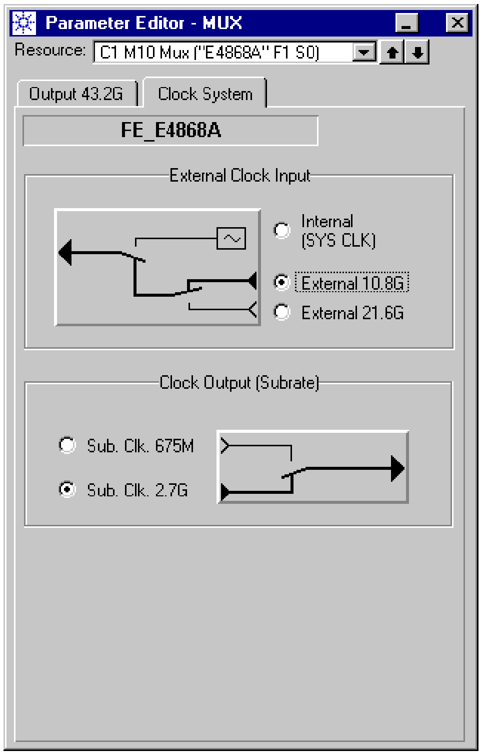
Note from the figure above that this is a feedback system with inherent delays.

The sampling phase of the multiplexer is determined by the PLL of the MUX module (blue).

The delays of the feedback signal path (red) must be compensated, so that the common phase of the 16 generated signals matches the sampling phase of the MUX module. A mismatch of Δt will occur, if, for example, the clock module is replaced.

Delay calibration allows you to shift the phase of the generated signals and thus to compensate for Δt.

5 Set the clock routing of the MUX module according to the figure above. This looks as shown in the figure below (see also *“How to Change the Clock Routing of a MUX Module” on page 218*).



Resource: C1 M10 Mux ("E4868A" F1 S0)

FE_E4868A

External Clock Input

○ Internal (SYS CLK)

● External 10.8G

○ External 21.6G

Clock Output (Subrate)

○ Sub. Clk. 675M

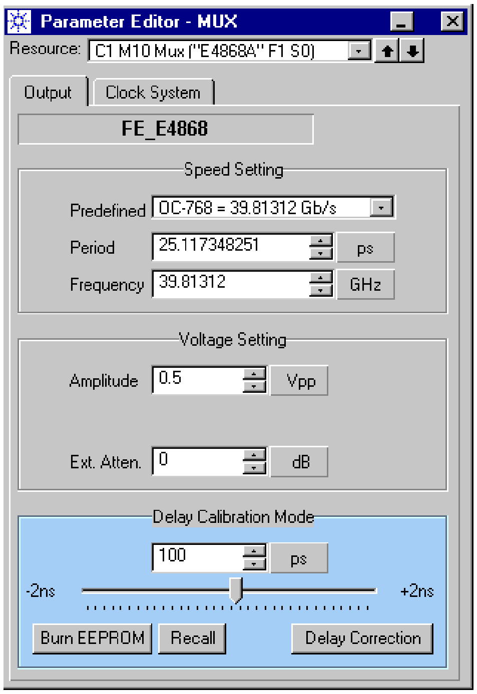
● Sub. Clk. 2.7G

Figure 294 **MUX Module Clock Routing**

6 Set the mode of the clock module to external source clock and measure the clock frequency (see *“How to Choose the Clock Source” on page 187*).

7 Connect the oscilloscope to the data output of the MUX module. By default (defined by the setting), the output data rate is set to 43.2 Gbit/s.

8 Open the Output Parameters page of the 43G MUX module (see also *"How to Change the Output Parameters of a MUX Module" on page 213*). Switch to *Delay Calibration Mode*.



Resource: C1 M10 Mux ("E4868A" F1 S0)

Predefined OC-768 = 39.81312 Gb/s

Period 25.117348251 ps

Frequency 39.81312 GHz

Amplitude 0.5 Vpp

Ext. Atten. 0 dB

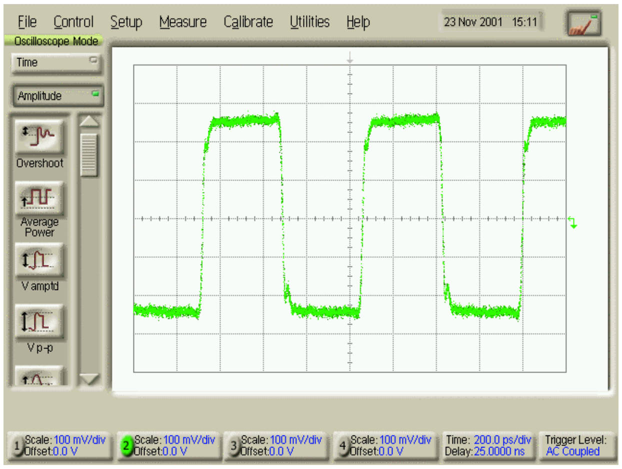
100 ps

Figure 295 MUX Module Output Parameters

The *Delay Calibration Mode* gives you a sweep range of ±2 ns (±2000 ps) for the delay vernier, as illustrated in the figure above. For a clock period of 25 ps (40 Gbit/s), this sweep range corresponds to ±80 periods.

9 Start the pattern generator and observe the oscilloscope.

You should see a pattern similar to the one shown below:



1 Scale: 100 mV/div Offset:0.0 V
2 Scale: 100 mV/div Offset:0.0 V
3 Scale: 100 mV/div Offset:0.0 V
4 Scale: 100 mV/div Offset:0.0 V
Time: 200.0 ps/div Delay: 25.0000 ns
Trigger Level: AC Coupled

Figure 296 Good Pattern

10 If the pattern is not as crisp and uniform as shown above, move the delay vernier slightly until it is.

If you move the delay vernier coarsely, you will find that areas of good and poor patterns alternate, as illustrated in the following figure.

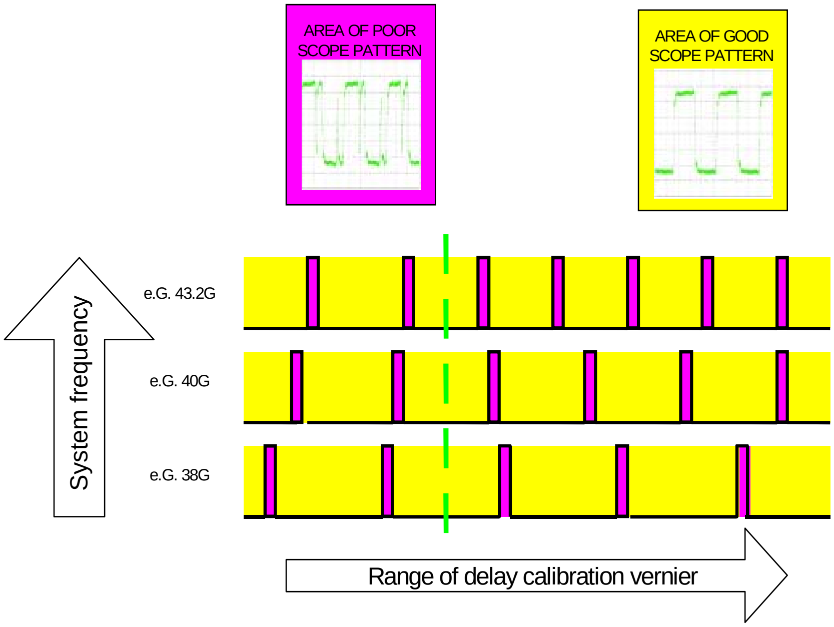

AREA OF POOR SCOPE PATTERN
AREA OF GOOD SCOPE PATTERN
System frequency
e.G. 43.2G
e.G. 40G
e.G. 38G
Range of delay calibration vernier

Figure 297 Alternating Scope Patterns

Distorted patterns indicate a mismatch between the sampling moment of the MUX module and the phase of the generated signals.

A good pattern indicates that the sampling point of the MUX module is within the common eye opening of the generated signals. The two scope patterns alternate, because the delay vernier allows you to shift the signal phase over many clock cycles. This is illustrated in the figure below:

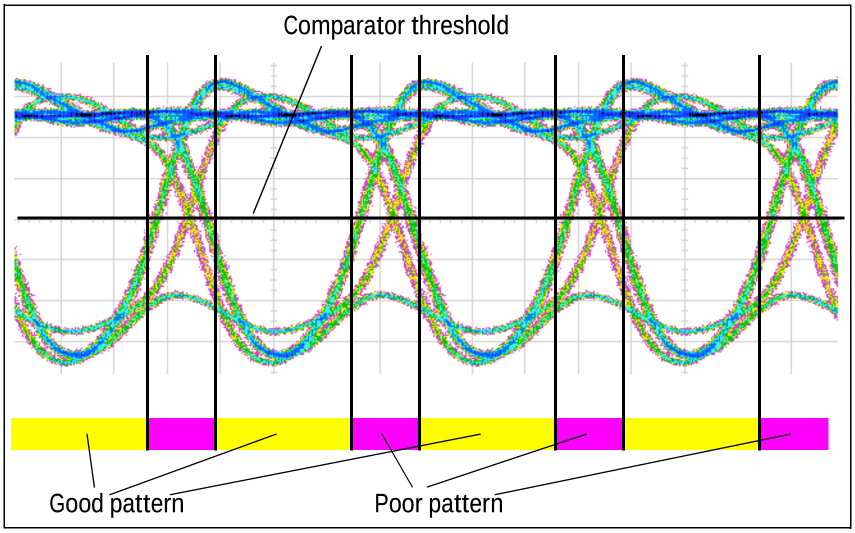


Figure 298 The Reason for Alternating Scope Patterns

NOTE

The ParBERT 43G pattern generator calibration uses the opposite principle than analyzer delay adjustment.

When you adjust the analyzer delay (which can be done manually or automatically) you move the analyzer sampling point over time until at least a suitable delay is found.

For the MUX module, the sampling point is given by the built-in PLL and fixed. When you calibrate the ParBERT 43G pattern generator, you shift the phase of the generated signals until the sampling point is in the middle of the eye opening.

This is simple and straightforward for one signal frequency. But you have to find a common delay that fits to the whole frequency range.

Calibration Procedure

The calibration procedure has three phases:

- Phase 1: Determine the optimum delay at 43.2 Gbit/s and 38 Gbit/s. Calculate the time difference.
- Phase 2: Increment the delay by 16 full periods and determine the optimum delay at 43.2 Gbit/s and 38 Gbit/s once more. Calculate this time difference, too.

- Phase 3: Compare the two time differences. Depending on the result, you are almost done, or you have to perform additional delay measurements.

The terms used for identifying the time measurements are illustrated in the figure below:

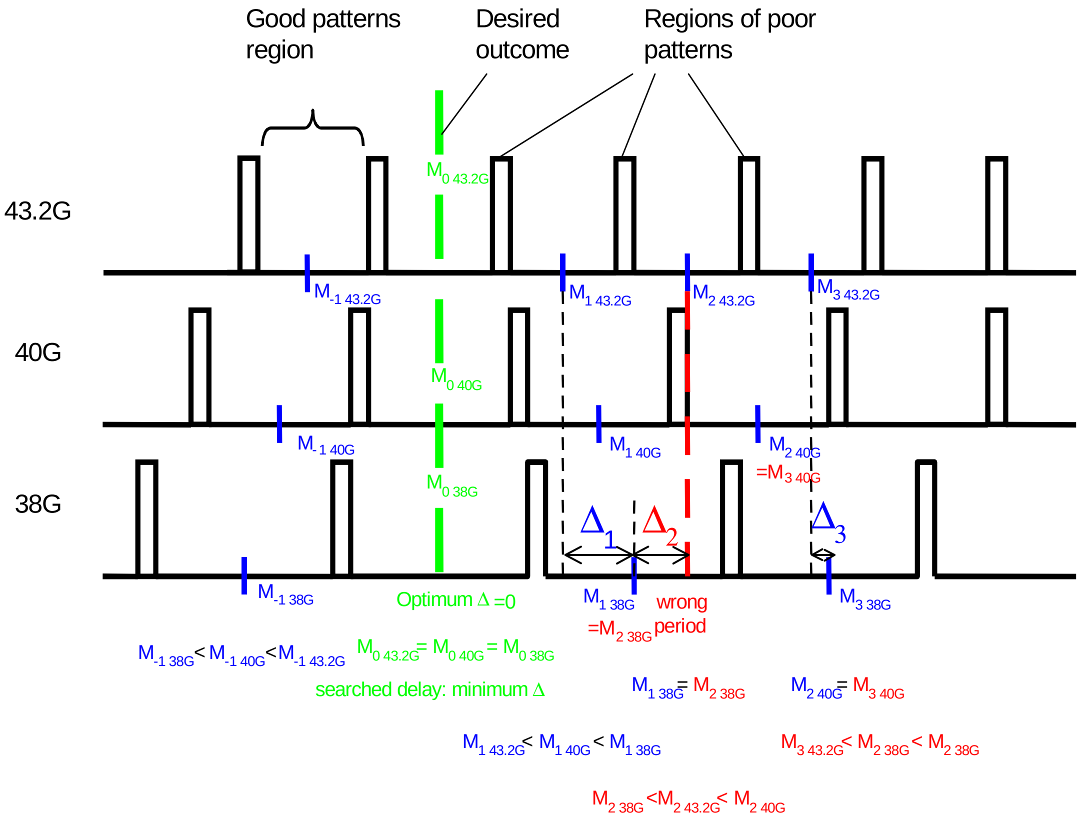


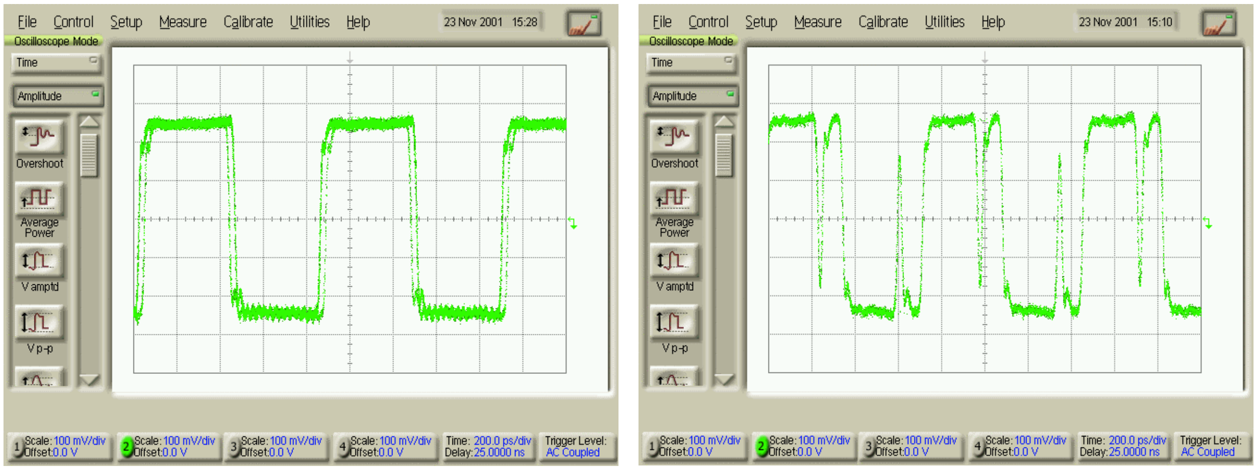
Figure 299 Calculated Values M1, M2, M3, and Deltas

Start the calibration with phase 1.

Phase 1 Procedure:

1 Using the delay vernier, increase the delay until the oscilloscope pattern just starts to deteriorate.

The left-hand picture of the following figure shows a good example of the beginning deterioration.



23 Nov 2001 15:28

Scale: 100 mV/div Offset:0.0 V | Scale: 100 mV/div Offset:0.0 V | Scale: 100 mV/div Offset:0.0 V | Scale: 100 mV/div Offset:0.0 V | Time: 200.0 ps/div Delay:25.0000 ns | Trigger Level: AC Coupled

23 Nov 2001 15:10

Scale: 100 mV/div Offset:0.0 V | Scale: 100 mV/div Offset:0.0 V | Scale: 100 mV/div Offset:0.0 V | Scale: 100 mV/div Offset:0.0 V | Time: 200.0 ps/div Delay:25.0000 ns | Trigger Level: AC Coupled

Figure 300 Poor and Very Poor Scope Patterns

2 Note the delay (in ps).

3 Decrease the delay until the oscilloscope pattern deteriorates again and note that delay, too.

4 Calculate the middle point as (1st delay + 2nd delay) / 2. This value is called $M_{i\ 43.2G}$. Start with i = 1. The result is $M_{1\ 43.2G}$.

5 Adjust the delay vernier to that position.

6 Set the data rate to 38 Gbit/s and repeat the steps 1 to 4. The result is $M_{1\ 38G}$.

7 Calculate the difference $\Delta_1 = M_{1\ 43.2G} - M_{1\ 38G}$.

Phase 2 Procedure:

1 Set the data rate back to 43.2 Gbit/s.

2 Increase the delay $M_{i\ 43.2G}$ by 370 ps.

3 Using the delay vernier and the oscilloscope, measure the width of the eye opening, and calculate $M_{i+1\ 43.2G}$.

4 Set the data rate to 38 Gbit/s and repeat step 3. The result is $M_{i+1\ 38G}$.

5 Calculate the difference Δ_{i+1}.

6 Compare the absolute values of Δ_i and Δ_{i+1} (see figure *“Calculated Values M1, M2, M3, and Deltas” on page 420*).

If Δ_i and Δ_{i+1} have the same sign and $|\Delta_i|$ is less than $|\Delta_{i+1}|$, then return to 43.2 Gbit/s, decrease the delay $M_{i\ 43.2G}$ by 370 ps, decrease i, and repeat the steps 3 to 5. The resulting mean values are $M_{0\ 43.2G}$ and $M_{0\ 38G}$.

Phase 3

The goal of phase 3 is to find a period with minimum deviation between the values of the delay M_i. This period has to be a “valid” period.

An invalid period is illustrated in the figure *“Calculated Values M1, M2, M3, and Deltas” on page 420*. There, the measurement of $M_{2\ 38G}$ falls into the same eye opening of the 38G-signal as the measurement of $M_{1\ 38G}$. This is not tolerable.

An invalid period can be identified by control measurements at 40 Gbit/s.

The whole procedure is illustrated in the following diagram:

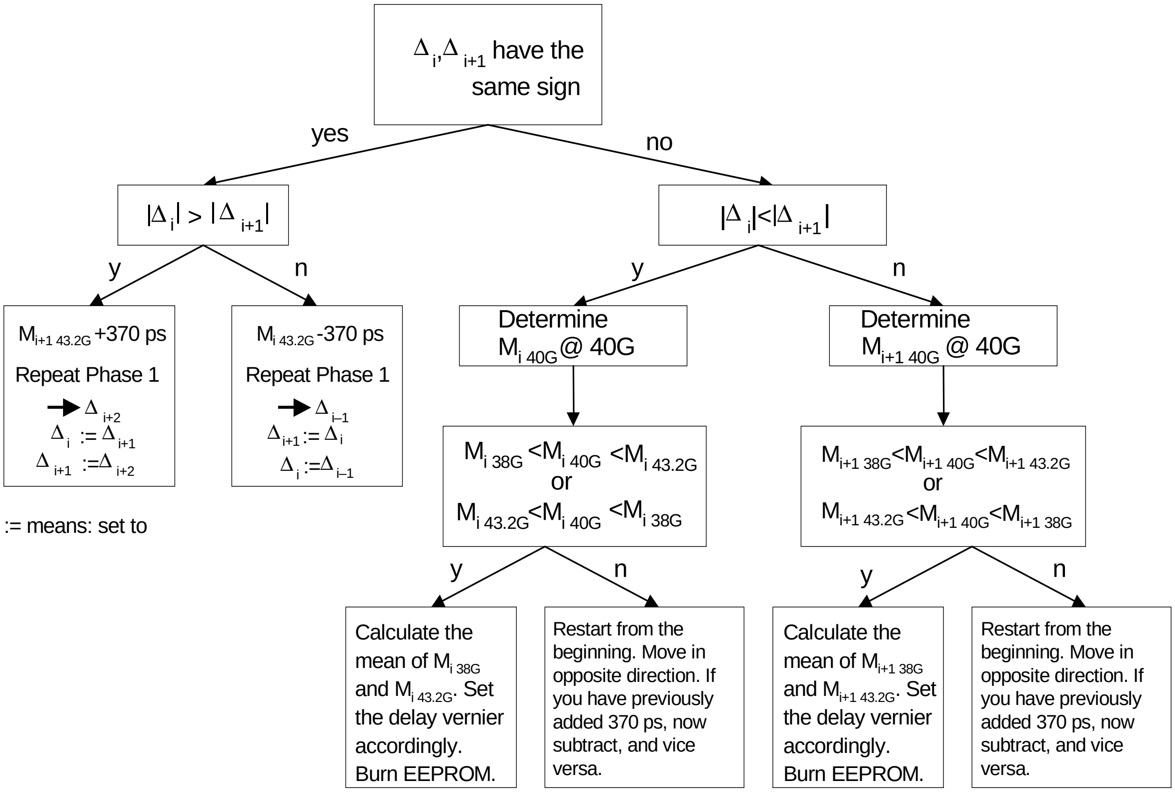

Δ_i,Δ_{i+1} have the same sign
yes
no
$|\Delta_i| > |\Delta_{i+1}|$
$|\Delta_i|<|\Delta_{i+1}|$
y
n
y
n
$M_{i+1\ 43.2G}$ +370 ps
Repeat Phase 1
$\rightarrow \Delta_{i+2}$
$\Delta_i := \Delta_{i+1}$
$\Delta_{i+1} := \Delta_{i+2}$
$M_{i\ 43.2G}$ -370 ps
Repeat Phase 1
$\rightarrow \Delta_{i-1}$
$\Delta_{i+1} := \Delta_i$
$\Delta_i := \Delta_{i-1}$
:= means: set to
Determine $M_{i\ 40G}$ @ 40G
Determine $M_{i+1\ 40G}$ @ 40G
$M_{i\ 38G} < M_{i\ 40G} < M_{i\ 43.2G}$
or
$M_{i\ 43.2G} < M_{i\ 40G} < M_{i\ 38G}$
$M_{i+1\ 38G} < M_{i+1\ 40G} < M_{i+1\ 43.2G}$
or
$M_{i+1\ 43.2G} < M_{i+1\ 40G} < M_{i+1\ 38G}$
y
n
y
n
Calculate the mean of $M_{i\ 38G}$ and $M_{i\ 43.2G}$. Set the delay vernier accordingly. Burn EEPROM.
Restart from the beginning. Move in opposite direction. If you have previously added 370 ps, now subtract, and vice versa.
Calculate the mean of $M_{i+1\ 38G}$ and $M_{i+1\ 43.2G}$. Set the delay vernier accordingly. Burn EEPROM.
Restart from the beginning. Move in opposite direction. If you have previously added 370 ps, now subtract, and vice versa.

Figure 301 Calibration Flow Chart

NOTE Remember that the calibration also has to be performed for the internal clock generated by the E4808A clock module, if the MUX module itself or cables between the data generator frontends and the MUX module have been replaced.

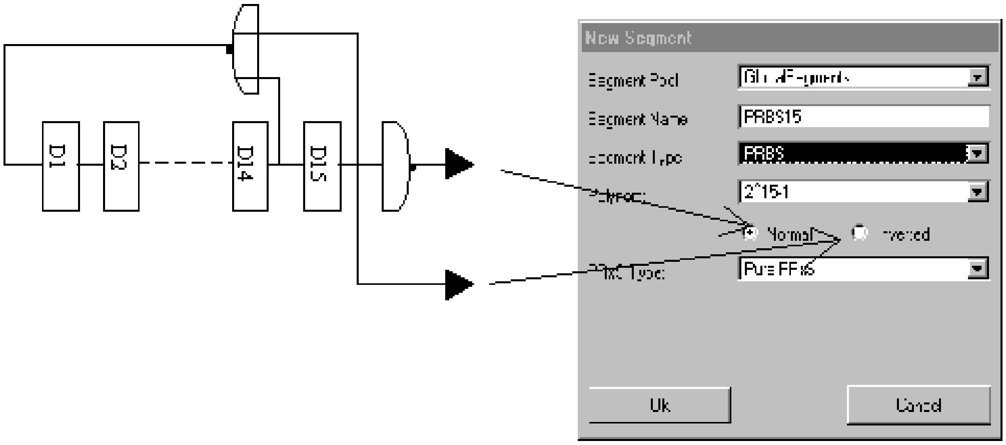
If this was the case, set up the MUX module and the clock module for the internal clock and repeat the calibration procedure.

Appendix B: PRBS / PRWS Data Segments

This appendix contains information how the Agilent 81250 Parallel Bit Error Ratio Tester generates Pseudo Random Bit/Word Streams (PRBS and PRWS). Furthermore, it describes the additional features available with PRBS/PRWS.

Pure and Distorted PRBS

The Agilent 81250 uses a shift register with appropriate feedback to generate the pseudo random data. An example is shown below.



Segment Pool: Global Segments
Segment Name: PRBS15
Segment Type: PRBS
Polynom: 2^15-1
● Normal ○ Inverted
PRBS Type: Pure PRBS

Figure 302 **The Generation of a 2^{15}–1 PRBS Using a Shift Register**

Here you can see, how an ITU-T O.151 compatible 2^{15}– 1 PRBS is generated.

If *Normal* is selected, the output of the inverter is used to generate the bit stream. The shift register starts with all flip-flops loaded to one. This causes the bit stream to start with the longest run of 15 zeros.

If *Inverted* is selected, the bit stream is sent to the DUT without the inverter. The shift register starts with all flip-flops loaded to one. This causes the bit stream to start with the longest run of 15 ones.

NOTE This implementation may be in contrast to other bit error rate testers. If you encounter problems when combining the Agilent 81250 with an undocumented PRBS generator or analyzer, you should try the inverted output.

The mathematical notation of the shift register with feedback taken from flip-flops 15 and 14 for the 2^{15}–1 PRBS is:

$$X^{15} + X^{14} + 1$$

A complete list of the available polynomials—using the mathematical notation—is given in the table below.

Table 22 PRBS Polynomials

PRBS	Polynomial	Comment
$2^5 - 1$	$X^5 + X^4 + X^2 + X^1 + 1$	
$2^6 - 1$	$X^6 + X^5 + X^3 + X^2 + 1$	
$2^7 - 1$	$X^7 + X^6 + 1$	Compatible with Agilent 70841A and 70845A. ITU-T V.29
$2^8 - 1$	$X^8 + X^7 + X^3 + X^2 + 1$	
$2^9 - 1$	$X^9 + X^5 + 1$	CCITT 0.153 / ITU-T V.52
$2^{10} - 1$	$X^{10} + X^7 + 1$	Compatible with Agilent 70841A and 70845A
$2^{11} - 1$	$X^{11} + X^9 + 1$	CCITT 0.152 / ITU-T 0.152
$2^{12} - 1$	$X^{12} + X^9 + X^8 + X^5 + 1$	
$2^{13} - 1$	$X^{13} + X^{12} + X^{10} + X^9 + 1$	
$2^{14} - 1$	$X^{14} + X^{13} + X^{10} + X^9 + 1$	
$2^{15} - 1$	$X^{15} + X^{14} + 1$	CCITT 0.151 / ITU-T 0.151
$2^{23} - 1$	$X^{23} + X^{18} + 1$	CCITT 0.151 / ITU-T 0.151
$2^{31} - 1$	$X^{31} + X^{28} + 1$	

All pure pseudo random bit/word streams are generated by hardware shift registers built into the data generator/analyzer modules.

Distorted PRBS are generated by simulating the shift register in the software and downloading the pattern into the channel memory. Due to memory constraints, only pure PRxS without any additional options is available for the two longest bit streams (2^{23}–1 and 2^{31}–1).

Variable Mark Density

The ratio of ones to zeros of a pure pseudo random bit stream is approximately 1 to 1. That means, 1/2 of all bits are ones. You can modify this ratio by using the option Variable Mark Density.

A PRBS with marker density 1/8, 1/4, 3/4, 7/8 generates bit streams with the respective ratio of ones to the total number of bits. For example, 3/4 means that on the average 3 out of four bits are ones.

The generation of such a 2^m-1 PRBS is done by combining a bit at position n with the bits generated m bits and 2*m bits later. The following formulas are used:

Table 23 PRBS Formulas

Mark density	Formula
1/8	PRBS[n] = PRBS[n] & PRBS [n+m] & PRBS [n+m*2]
1/4	PRBS[n] = PRBS[n] & PRBS [n+m]
3/4	PRBS[n] = PRBS[n] | PRBS [n+m]
7/8	PRBS[n] = PRBS[n] | PRBS [n+m] | PRBS [n+m*2]

Extended Zeros / Ones

Normal Mode

A pure pseudo random bit stream in **normal** mode generates the longest run of zeros as the first bits. Thus, for a $2^m - 1$ PRBS the first m bits are zero. With the option Extended Zeros you can extend this sequence of m zeros.

Starting at bit position m+1 in the pattern memory a specified number of bits is set to zero. The following bit is then forced to be a one.

Inverted Mode

The opposite applies for an **inverted** PRBS. Here you can extend the number of ones that are sent as the first bits.

Principle of Operation

If ones shall be extended to a normal PRBS, the start phase of the PRBS is modified in such a way that the longest run of ones ends exactly at bit position m (first bit is zero, bit one to m are ones). Therefore, the insert position of the extending ones is again at index m+1.

The same applies for an inverted PRBS that shall be extended with zeros.

Error Insertion

To produce a distinct bit error rate in your test sequence, you can use the Error Insertion option. This option inserts errors into the pseudo random bit stream.

This is achieved by filling a pure PRBS stream into memory and then toggling as many bits as specified at random positions.

For example, 2 errors inserted into a $2^{15}-1$ PRBS yields an error rate of

$$2 / (2^{15}-1) = 6.1037e^{-5}.$$

Pure and Distorted PRWS

A Pseudo Random **W**ord Stream (PR**W**S) is used to send pseudo random data to a multiplexer (MUX) input port. PRWS is also used to specify the data expected from a demultiplexer (DEMUX) output port.

Pure PRWS

If the signals are connected to the MUX input pins in correct order, a PRBS appears at the output pin of the MUX. This is illustrated in the figure below. The numbers in the figure represent the nth bit of a PRBS data stream.

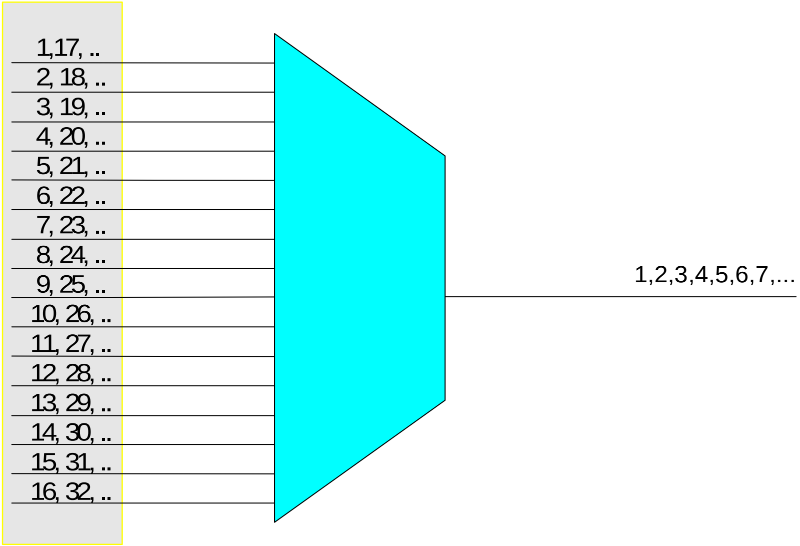


Figure 303 Sending a PRWS to a 16 : 1 MUX

If the MUX-width is a number in power of two, then the individual input data streams to the MUX are PRBS streams of the same polynomial as the resulting serial data stream at the MUX output. For a MUX with a width other than a power of two, PRBS using different polynomials must be provided to the MUX. This is done automatically by the software.

Restrictions

However, there are a few invalid combinations of PRBS-lengths and MUX-widths. They are listed in the table below.

Table 24 Known Exceptions for MUX-Widths up to 256

PRWS	Invalid MUX-widths
$2^5 - 1$	31, 62, 93, 124, 155, 186, 217, 248
$2^6 - 1$	3, 7, 9, 18, 21, 27, 36, 42, 45, 54, 63, 72, 81, 84, 90, 99, 105, 108, 117, 126, 135, 144, 147, 153, 162, 168, 171, 180, 189, 198, 207, 210, 216, 225, 231, 234, 243, 252
$2^7 - 1$	127, 254
$2^8 - 1$	3, 5, 15, 17, 34, 51, 68, 85, 102, 119, 136, 153, 170, 187, 204, 221, 238, 255
$2^9 - 1$	7, 73, 146, 219
$2^{10} - 1$	3, 11, 31, 33, 66, 93, 99, 132, 165, 198, 231
$2^{11} - 1$	23, 89
$2^{12} - 1$	3, 5, 7, 9, 13, 15, 21, 35, 39, 45, 63, 65, 91, 105, 117, 130, 195
$2^{13} - 1$	No exceptions
$2^{14} - 1$	3, 43, 86, 127, 129, 172, 215
$2^{15} - 1$	7, 31, 151, 217
$2^{23} - 1$	47
$2^{31} - 1$	No exceptions

Some exceptions are obvious: For example, a 2^5-1 PRBS generates a bit stream that is repeated every 31 bits. A multiplexer with 31 inputs would then require constant signals at the input pins to provide a PRBS at the serial output. But as the pure PRBS is generated using a shift register, it is mandatory that the individual inputs to the MUX itself are also PRBS streams of a certain polynomial.

If you need to have a PRWS for a MUX-width that is listed in the exception table, use this workaround: Create a non-pure PRWS. For example, specify it as a PRWS with a variable marker density 1/2.

Distorted PRWS

The different types of non-pure Pseudo Random Words Streams (PRWS) are generated in a similar way as the various PRBS types. The PRWS generation is done in two steps:

1. A PRBS with the specified parameters is generated.
2. The data memories of the related channels are filled in such a way that at the serial side of the MUX the desired PRBS appears.

Appendix C: Glossary

This appendix provides a glossary of the terms used in the user interface and in the documentation.

10GbE: 10 Gbit/s Ethernet

Action upon an event: Any combination of the following:

- Go to sequence block.
- Set the TRIGGER OUTPUT of the master clock module.
- Set the VXI ECL T0/T1 trigger lines.

ActiveX controls: Reusable software objects that can be pasted into many programming environments.

Agilent 81200 pnp driver: A plug and play driver for the 81200 platform based on the VISA standard.

Analog channel addition: Voltage addition of two 675 MHz signal generator channels of one module. Requires at least one E4838A frontend.

API: Application Programming Interface.

Automatic Bit Synchronization: Analyzer sampling phase adjustment based on measuring the BER. Sets and shows the phase delay with respect to the analyzer clock.

Automatic Delay Alignment: Analyzer sampling delay optimization if a time window is known. Sets and shows the full delay since start.

BER: Bit Error Rate. The number of errored bits divided by the number of received bits.

BIOS: Basic I/O System—the microprocessor programs loaded into the EEPROMs of the VXI modules.

Block: A portion of a test sequence which references segments that define generated and expected data. A block refers to all data ports. Its length has to be a multiple of the segment resolution. Single blocks and groups of blocks can be repeated (loops). A trigger pulse can be issued at the beginning of a block. Actions upon events can be associated with a block.

Cable delay compensation A procedure that compensates for signal delays between the frontend connectors and the DUT board. Ensures that all signals arrive phase-synchronized, at the DUT board as well as at the analyzers. Cable delay compensation can be run from the Deskew Editor.

Capture Data A measurement mode. In capture mode, the analyzer frontends capture data until the sequence expires or their memory is filled.

Channel: The circuitry behind a frontend connector, which includes data generating or analyzing capabilities, data memory, frequency multiplexing, and so on.

CLOCK / REF INPUT: A connector of the clock module. It allows to connect an external clock source.

Clockgroup: The sum of modules connected to a single clock module.

ClockgroupNumber: Identifies the clock master (= 1) and up to two slaves (2 and 3).

Compare and Acquire Around Error A measurement mode. The Compare and Acquire around Error mode compares and acquires data in real time. If an error occurs, it is possible to define when the system should stop after the occurrence of the error.

Compare and Capture A measurement mode. The Compare and Capture mode compares and acquires data in real time. It continues until the sequence expires or the Stop button is pressed.

Connector: An output or input connector of a frontend. Differential connectors are treated as one connector.

ConnectorNumber: Identifies the connector of a module and is counted from module top to bottom (1 to 4). Differential connectors are counted as one connector.

Data port: A DUT port that receives or returns digital data, such as a data or address bus.

DCE: Data Communication Equipment.

Delay Auto Calibration A procedure that has to be run after adding or replacing frontends or modules. Can be started from the *System* menu.

Delay vernier: A slider provided by the Parameter Editor for analyzer frontends plugged into an E4861A or E4832A data generator/analyzer module. Allows to move the sampling point while a test is running.

Deskew Editor A ParBERT tool for phase-synchronizing the frontend connectors and compensating for signal delays caused by the test setup. Can be started from the *Go* menu.

Digital channel addition: Exclusive OR (modulo 2) addition of two or four signal generator channels of one module.

DSC DeSkew Channel. The 17th channel used by SFI-5 applications for deskewing the 16 data channels.

DSR: Digital Stimulus/Response. A general term used for characterizing instruments that source digital data to a device and analyze the digital response.

DSRA: Default name of the basic ParBERT system with the first clock module. Additional independent systems are labeled in ascending order: DSRB, DSRC, etc.

Error State Display: Shows captured data and errors in tabular form.

Event: A signal that causes an action.

Event causing deferred action: An event that causes an action at the end of the currently executed sequence block.

Event causing immediate action: An event that is serviced as fast as possible, without waiting for the end of the currently executed sequence block.

EXT INPUT: A connector of the clock module. It allows to start/stop the system by an external signal.

Fast Bit Synchronization: A method to align expected PRxS data with incoming PRxS data. Does not change the analyzer sampling delay.

FM factor: Frequency Multiplier factor. The individual factor by which a channel frequency differs from the system clock frequency. Choices are restricted by the FMR.

FMR: Frequency Multiplier Range. The available factors for multiplying the system clock frequency. The actual range depends on the segment resolution and the type of module.

Frontend: Generator or analyzer plug-in of a data module.

Handle: The identification of a system, such as DSRA, in SCPI commands.

Master clock module: The E4805B or E4808A clock module that controls clockgroup #1 of a system. It can additionally control up to two slave clock modules (clockgroup #2 and #3).

Module: One of the following:

- E4805B Clock Module
- E4808A Clock Module
- E4832A Data Generator/Analyzer Module (up to 675 Mbit/s)
- E4861A Data Generator/Analyzer Module (up to 2.7 Gbit/s)
- E4861B Data Generator/Analyzer Module (up to 3.35 Gbit/s)
- E4868A/B 43.2 Gbit/s Multiplexer (MUX) Module
- E4869A/B 43.2 Gbit/s Demultiplexer (DEMUX) Module
- E4866A 10.8 Gbit/s Data Generator Module
- E4867A 10.8 Gbit/s Data Analyzer Module
- N4868A 10.8 Gbit/s Booster Module
- E4810A 3.35 Gbit/s optical-electrical Data Generator Module
- E4811A 3.35 Gbit/s optical-electrical Data Analyzer Module

ModuleNumber: Identifies the module within a clockgroup (range 1 to 11).

MUX / DEMUX: Multiplexer/Demultiplexer. Also used for identifying the E4868A multiplexer and the E4869A demultiplexer modules.

OC: Optical Carrier. Describes the optical characteristics of a SONET. OC-1 corresponds to STS-1, OC-768 to STS-768.

Parameter Editor: A versatile component of the ParBERT user software that allows to set the characteristics of the clock module, the DUT data ports, and selected ParBERT channels.

PNP: 81200 Plug and Play peripheral driver for VXI components.

Port: A group of DUT input or output pins with identical or similar properties. There are data ports and pulse ports. Data ports are divided into DUT input ports and DUT output ports.

PRBS / PRWS: Pseudo Random Bit/Word data Stream.

Pulse port: A DUT port that receives parametric signals, such as a clock pulse.

SCPI: Standard Command language for Programmable Instruments.

SDH: Synchronous Digital Hierarchy.

Segment resolution: The length of a data word in the module memory. Range: 1 to 16 bits for E4832A modules, 16 to 64 bits for E4861A modules, 1 to 128 bits for E4861B modules. The minimum segment resolution depends on the chosen system clock frequency.

Segment: Contains the data to be generated or expected. Segment types are: Memory, PRBS, PRWS, or SFI5. Type memory is a pattern that consists of vectors and traces. PRxS means algorithmic data. SFI5 refers to a 16-bit wide PRWS plus the DSC.

Sequence: The overall stream of generated and expected data, formed by sequence blocks.

Sequence Editor: Three ParBERT tools for creating and maintaining the sequence: Standard Mode Sequence Editor, Detail Mode Sequence Editor, and Data/Sequence Editor. Can be started from the *Go* menu.

SERDES: Serializer/Deserializer. Multiplexers/demultiplexers used in SONET applications.

Setting: The complete setup for a DUT test, including all parameters and references to the test patterns (segments).

SFI-5: SERDES Framer Interface, spec #5.

SONET: Synchronous Optical Network. A standard from Bellcore and ITU-T (formerly CCITT).

Start delay: Analyzer sampling point setting with the Parameter Editor. Describes also the delayed start of generators after starting a test.

STM: Synchronous Transfer Mode. STM-1 is the base level of the SDH and refers to a data rate of 155.52 Mbit/s. This corresponds to STS-3.

STS: Synchronous Transport Signal. STS-1 is the base level of SONET and refers to a data rate of 51.84 Mbit/s.

System: An independent ParBERT system consists of one master clock module and at least one data generator/analyzer module. Several systems may share a common VXI frame.

Terminal: A signal line (a DUT pin) assigned to a port.

Trace: Specifies the serial data transmitted to or expected from a terminal.

Trigger Pod: An option of the E4805B or E4808A master clock module for detecting external events.

Vector: Specifies all the parallel, simultaneous bits of a port within a segment.

VISA: Virtual Instrument Software Architecture—a common standard of functional calls for controlling VXI-based instruments.

VXI: VME-bus eXtension for Instrumentation—a bus system for building modular instruments.

Waveform Viewer: Shows generated and captured data as well as errors in graphical form. Allows to investigate phase relationships.

XAUI: 10 Gbit/s Ethernet (X = 10GbE) Attachment Unit Interface. A 4 x 3.125 Gbit/s parallel interface between the MAC (media access control) sublayer and the PHY (physical layer Tx/Rx), specified in IEEE 802.3ae.

Xenpak: A multisource agreement (MSA) for the XAUI <=> PHY package.

Zero adjust: A procedure that compensates for internal delays between the different data modules and frontends. Ensures that all generators and analyzers are phase-synchronized. Zero adjust can be run from the Deskew Editor.

Index

Printed in Germany May 2003

5988-4887EN

Agilent Technologies