# 2108 Development System v2.1

# User Manual

August 28, 2007



**EADS North America Defense Test and Services, Inc.**
P.O. Box 19541 Irvine, CA 92623-9541 4 Goodyear Irvine, CA 92618-2002
T: (949) 859-8999 (800) 722-2528 F: (949) 859-7139 www.eads-nadefense.com

## LICENSE AGREEMENT

This EULA is a binding legal agreement between you and EADS North America Defense Test & Services, Inc., 4 Goodyear Street, Irvine, CA 92618 (hereinafter "Licensor") for the materials accompanying this EULA, including the accompanying computer software, any associated media, any printed materials and any "online" or electronic documentation (hereinafter the "Software"). By installing the Software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or attempt to use the Software.

### 1. Grant of License

The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold. This EULA grants you the following rights:

A. You may install, use, access, display and run only one copy of the Software, on a single computer, workstation or terminal ("Computer"). The primary user of the Computer on which the Software is installed may make a second copy for his or her exclusive use for archival purposes only.
B. Your license rights under this EULA are non-exclusive. All rights not expressly granted herein are reserved by Licensor.
C. You may not sell, transfer or convey the Software to any third party without Licensor's prior express written consent.

### 2. Replacement, Modification and/or Upgrades

Licensor may, from time to time, and for a fee, replace, modify or upgrade the Software. When accepted by you, any such replacement or modified Software code or upgrade to the Software will be considered part of the Software and subject to the terms of this EULA (unless this EULA is superseded by a further EULA accompanying such replacement or modified version of or upgrade to the Software).

### 3. Termination

You may terminate this EULA at any time by destroying all your copies of the Software. Your license to the Software automatically terminates if you fail to comply with the terms of this agreement. Upon termination, you are required to remove the Software from your computer and destroy any copies of the Software in your possession.

### 4. Copyright

A. All title and copyrights in and to the Software (including but not limited to any images, photographs, animations, video, audio, music and text incorporated into the Software), any accompanying printed materials, and any copies of the Software, are owned by Licensor or its suppliers. This EULA grants you no rights to use such content. If this Software contains documentation that is provided only in electronic form, you may print one copy of such electronic documentation. Except for any copies of this EULA, you may not copy the printed materials accompanying the Software.

B. You may not reverse engineer, de-compile, disassemble, alter, duplicate, modify, rent, lease, loan, sublicense, make copies of, create derivative works from, distribute or provide others with the Software in whole or part, transmit or communicate the application over a network.

### 5. Export Restrictions

You may not export, ship, transmit or re-export software in violation of any applicable law or regulation including but not limited to Export Administration Regulations issued by the U. S. Department of Commerce.

### 6. Disclaimer of Warranties

LICENSOR AND ITS SUPPLIERS PROVIDE THE SOFTWARE "AS IS" AND WITH ALL FAULTS, AND HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO ANY (IF ANY) IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF LACK OF VIRUSES, AND OF LACK OF NEGLIGENCE OR LACK OF WORKMANLIKE EFFORT. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, OF QUIET ENJOYMENT, OR OF NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SOFTWARE IS WITH YOU.

### 7. Limitation of Damages

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR OR ITS SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SOFTWARE AND WHETHER BASED ON CONTRACT, TORT, NEGLIGENCE, STRICT LIABILITY OR OTHERWISE, EVEN IF LICENSOR OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS EXCLUSION OF DAMAGES WILL BE EFFECTIVE EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE.

### 8. Arbitration

Any dispute arising under this EULA will be subject to binding arbitration by a single Arbitrator with the American Arbitration Association (AAA), in accordance with its relevant industry rules, if any. The parties agree that this EULA will be governed by and construed and interpreted in accordance with the laws of the State of California. The arbitration will be held in Irvine, California. The Arbitrator will have the authority to grant injunctive relief and specific performance to enforce the terms of this EULA. Judgment on any award rendered by the Arbitrator may be entered in any Court of competent jurisdiction.

### 9. Severability

If any term of this EULA is found to be unenforceable or contrary to law, it will be modified to the least extent necessary to make it enforceable, and the remaining portions of this Agreement will remain in full force and effect.

### 10. No Waiver

No waiver of any right under this EULA will be deemed effective unless contained in writing signed by a duly authorized representative of the party against whom the waiver is to be asserted, and no waiver of any past or present right arising from any breach or failure to perform will be deemed to be a waiver of any future rights arising out of this EULA.

### 11. Entire Agreement

This EULA constitutes the entire agreement between the parties with respect to its subject matter, and supersedes all prior agreements, proposals, negotiations, representations or communications relating to the subject matter. Both parties acknowledge that they have not been induced to enter into this EULA by any representations or promises not specifically stated herein.

# Table of Contents

# 1. Introduction

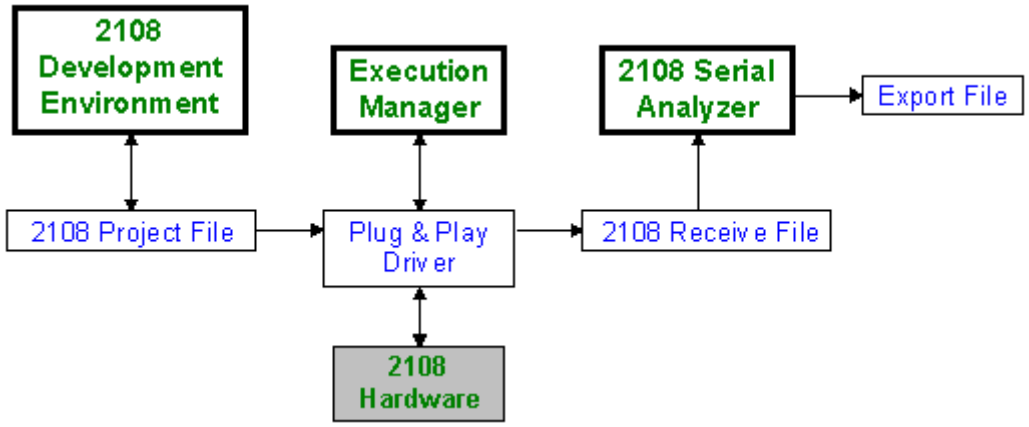
The Talon 2108 is a modern VXI-based high-performance instrument that is used as a serial word generator, serial logic analyzer or as a key component in a automated test system. The 2108 is an extremely adaptable instrument that gets much of its versatility through software control. The 2108 Development System is a collection of software components that provide a comprehensive approach for developing, debugging and maintaining applications for the Talon 2108. Tasks from design to deployment can be accomplished without writing a single line of code.

Coupled with the Talon 2108 hardware, the 2108 Development System provides an application developer with a modern, graphical, open-architecture tool that is applicable to a broad range of operating situations and environments.

## 1.1 System Architecture

The 2108 Development System includes the 2108 Development Environment, Execution Manager, Serial Analyzer, Plug&Play Driver DLL and 2108 project file.

Serial test data is read and written to a project file by the 2108 Development Environment. From there it is loaded and executed on the 2108 hardware by the Execution Manager. Alternately, other applications may use the Plug&Play Driver DLL to directly control loading and test execution on the 2108. Recorded receive data is formatted and displayed by the 2108 Serial Analyzer.




The 2108 Development Environment is used to visually define and view serial transmit and recording information for the Talon 2108. Its advanced user interface design streamlines the process of creating serial patterns and record sequences. This 32-bit Windows application defines serial test data and timing without cumbersome and arcane programming. In addition, it provides graphical access to the 2108 hardware controls. The 2108 Development Environment writes and reads a project file which stores 2108 application data.

The 2108 Development Environment may be used on a workstation independent of the 2108 hardware. Alternatively, integrated operation with a 2108 is supported when used with the Plug&Play Driver DLL and the Execution Manager.

The Execution Manager is a Windows application that uses the Plug&Play Driver DLL to handle loading and executing a project file on a 2108. Test results are displayed and several execution options are supported. The Execution Manager writes record sequence results to a binary file. This binary file is subsequently read and displayed by the 2108 Serial Analyzer.

The 2108 Serial Analyzer graphically displays the data recorded by a 2108 Receive channel. The receive data is overlaid with user-defined display templates. The display templates define decoding and formatting that is applied to the “raw” recorded data. The data and display formatting can be exported to several spreadsheet-compatible formats (e.g. comma delimited or tab delimited).

The 2108 Plug&Play Driver DLL provides high-level software access to the 2108 hardware. It coordinates the loading and execution of a project file on the 2108. It can be used to integrate 2108 control with other instruments or applications. Its programming interface supports development in a wide variety of languages and environments. These include HP-VEE, National Instruments LabWindows CVI, National Instruments LabView, C++, Java and Visual Basic.

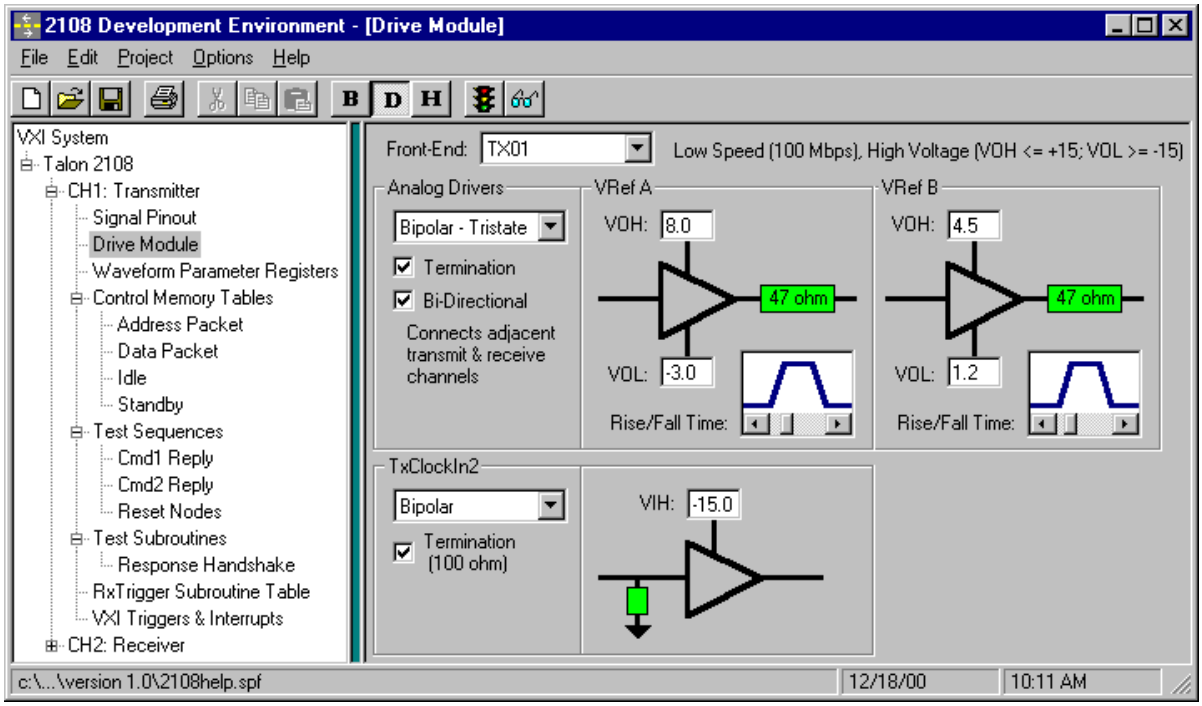
The 2108 project file is a repository for 2108 software control settings, timing and serial data. It holds all the information required for a specific operation. This greatly simplifies project development, tracking and maintenance. 2108 project files are identifiable by their file extension (*.SPF).

## 1.2 2108 Hardware

The Talon 2108 is a VXI-based instrument that is used as a serial word generator, serial logic analyzer or both. Its C-sized chassis houses a motherboard that accommodates up to four separate serial channels. These channels can be transmitters, receivers or a combination. Each channel has a separate front-end module that controls clock rates and voltage ranges. Different front-ends are available to meet various interface requirements.

# 2. 2108 Development Environment

The 2108 Development Environment (2108DevEnv) manages the creation, editing and viewing of a 2108 project file. A 2108 project file defines the operating parameters of one or more serial transmitter, or receiver, channels contained within a 2108 instrument module. The 2108DevEnv is also used to dispatch companion tools and coordinate import and conversion tasks.



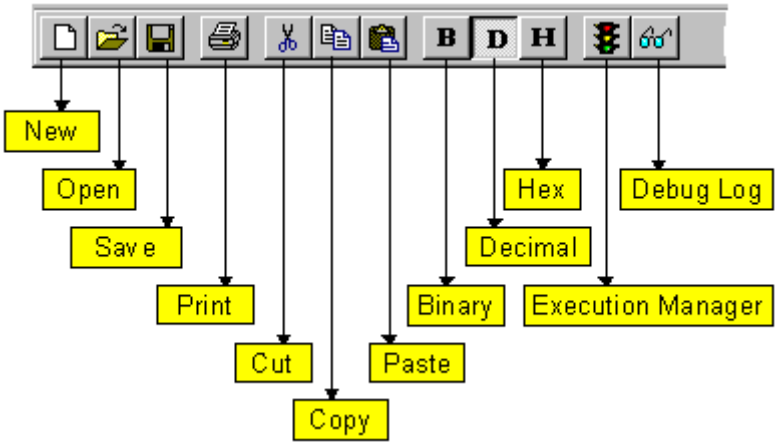

On the left side of the 2108DevEnv is a project list which is an outline or tree view of the elements contained in the project. Selecting an element causes it to be displayed to the right of the project list. A vertical green bar adjusts the width of the project list by click-dragging to the left or right. A single left mouse click on a branch expands [+] or collapses it [-]. Scroll bars are automatically provided if the list exceeds the size of the pane. Elements are added with the Project Menu or a similar pop-up menu (right mouse click on project list). Placement of new elements is relative to the currently selected item. Note that elements with user defined names are listed in alphabetical order.

A 2108 project contains transmit and/or receive channels and their associated settings and parameters. A transmit channel has two or more Control Memory Tables (CMTs) and one or more Test Sequences. The other elements of a transmit channel are fixed at one each. A receive channel has five elements that define its behavior.

The toolbar near the top of the 2108DevEnv provides a quick way to access commonly used commands. All of these commands are also available via menus and many are assigned shortcut keys. A status bar at the bottom of the 2108DevEnv displays the name of the current project file.

## 2.1 Toolbar

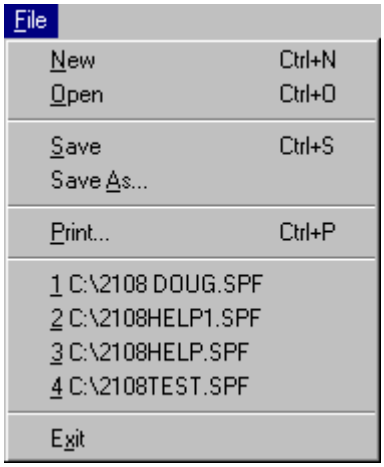
The toolbar on the 2108 Development Environment provides quick access to commonly used commands. All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function.



# 2.2 Menus

## 2.2.1 File Menu

The 2108 Development Environment File Menu is used to manage the loading and saving of 2108 Project files (*.SPF). With this menu, 2108 project files are created, loaded, saved and renamed. The print command sends a screen capture of the application to the selected printer. A file history list permits quick reloading of recently accessed project files. The 2108 Development Environment can also be closed from this menu.



| Menu Option | Description |
|---|---|
| **New** | Opens a file browser for naming and placing a new project file. A default configuration is displayed in the project list. |
| **Open…** | Opens a file browser for choosing a project file. The chosen file is loaded and displayed in the project list. |
| **Save** | Updates the project file with the latest editing changes. |
| **Save As…** | Creates a new project file with the latest editing changes. It then becomes the current project file. |
| **Print…** | Prints a screen capture of the application on the selected printer. |
| **File History** | This provides a quick selection from the last four project files loaded. The list is updated each time a new file is read. The file names are stored in a state file (2108DevEnv.ini). |
| **Exit** | Closes the 2108 Development Environment. If the current project has not been saved, the user is prompted to do so before closing. |

## 2.2.2 Edit Menu

The 2108 Development Environment Edit Menu supports editing operations for the Control Memory Table and Test Sequence windows. Editing these windows primarily involves operations on rows in a grid. The rows can be cut, copied, pasted and deleted. Note that a paste command is ignored if the system Clipboard contains the wrong type of data.

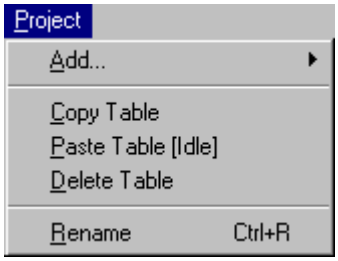
Individual rows are selected by a left mouse click in the leftmost column. Multiple rows are selected by holding the left mouse button down and dragging down the leftmost column.

Several toolbar buttons and shortcut keys are provided to more easily initiate these operations. This menu is also accessible as a pop-up menu (right mouse click on the grid). Individual fields, and controls on other windows, always support the shortcut keys for cut (**Ctrl+X**), copy (**Ctrl+C**) and paste (**Ctrl+V**).


| Menu Option | Description |
|---|---|
| **Cut** | Copy the selected rows to the system Clipboard and remove them from the control. The remaining rows are shifted up. |
| **Copy** | Copy the selected rows to the system Clipboard. |
| **Paste** | Insert the contents of the system Clipboard into the grid, starting at the topmost selected row. |
| **Delete** | Delete one or more selected rows. A row must be fully selected in order for it to be deleted. |
| **Insert** | Insert a new row above the selected row in the grid. |

## 2.2.3 Project Menu

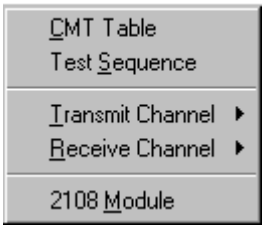
The 2108 Development Environment Project Menu is used to manage project contents. It controls editing, adding, deleting and renaming of items in the project list. It can also be accessed as a pop-up menu (right mouse click on project list). Some menu options are dependent upon which project element is selected. For example, Signal Pinout entries cannot be independently copied, deleted or renamed.




| Menu Option | Description |
|---|---|
| **Add…** | Presents a submenu of elements that can be added to the project list. |
| **Copy** | Copy the selected item in the project list to the paste buffer. A copy of a transmit channel includes its CMTs and test sequences. |
| **Paste** | Insert the contents of the paste buffer relative to the selected item in the project list. This option is disabled when there is nothing in the paste buffer. |
| **Delete** | Removes the selected item from the project list. Some elements cannot be individually removed. |
| **Rename** | Allows the currently selected CMT, or test sequence, to be renamed. Also accomplished by pressing **Ctrl+R** or with a double left click. |

## 2.2.4 Add Menu

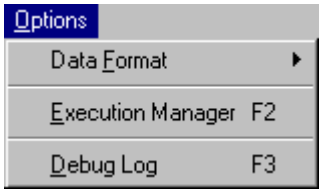
The Add Menu, a submenu to the 2108 Development Environment Project Menu, is used to add elements to the project list. Placement of new elements is relative to the currently selected item. CMTs and test sequences are always listed in alphabetical order. Items on the Add Menu are only enabled if they relate to the currently selected element on the project list. When adding a Transmit or Receive channel, a further submenu is presented for selecting the channel number to add.




| Menu Option | Description |
|---|---|
| **CMT Table** | Add a new CMT to the currently selected Control Memory Table hierarchy. |
| **Test Sequence** | Add a new test sequence to the currently selected Test Sequence hierarchy. |
| **Transmit Channel** | Add a new Transmit channel to the currently selected Talon 2108 module. |
| **Receive Channel** | Add a new Receive channel to the currently selected Talon 2108 module. |
| **2108 Module** | Add a new Talon 2108 module to the bottom of the project list. |

## 2.2.5 Options Menu

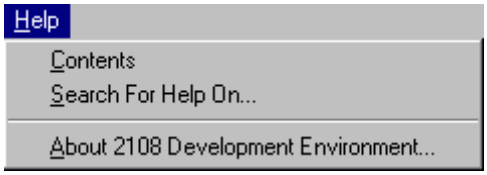
The 2108 Development Environment Options Menu is used to access display options, external utilities and logging information. The Execution Manager is an external utility for interactive control of a 2108. The Debug Log is an auxiliary window for displaying operational warnings and errors. They both can be initiated from the Options Menu or toolbar.




| Menu Option | Description |
|---|---|
| **Data Format** | Presents a submenu of data display formats (Binary, Decimal, Hexadecimal). Also available from the toolbar. |
| **Execution Manager** | Starts the Execution Manager utility for interactive control of a 2108. The Execution Manager automatically reads the current project file and writes it to the 2108. Function key **F2** also performs this task. |
| **Debug Log** | Allows the user to toggle between the Debug Log window and the 2108DevSys. Function key **F3** also performs this task. |

## 2.2.6 Help Menu

The 2108 Development Environment Help Menu provides access to the 2108 Development System help file and version information. The help file is most easily accessed by pressing the **F1** function key. Using **F1** activates the help file topic that relates to the current window or dialog box.

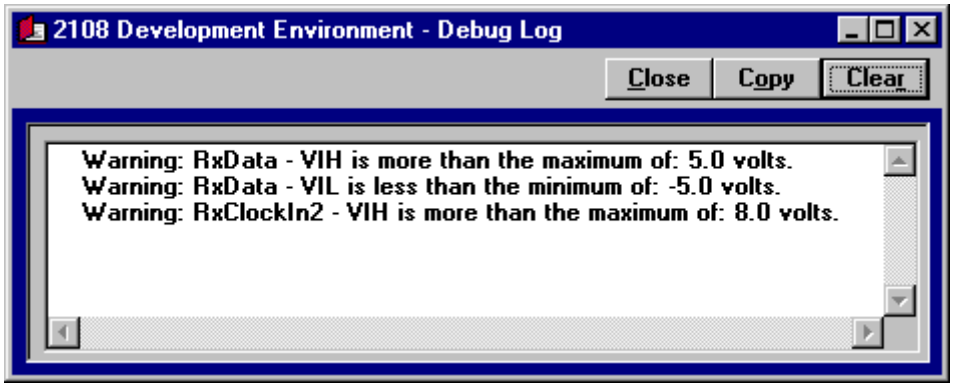



| Menu Option | Description |
|---|---|
| **Contents** | Displays the Contents page of the help file. Function key **F1** displays context-sensitive help for the current window or dialog. |
| **Search for Help On…** | Displays the Search index of the help file. Use this to locate specific information in the help file. |
| **About 2108 Development Environment** | Displays a dialog containing version information and a copyright notice. |

## 2.3 Dialogs & Windows

### 2.3.1 Debug Log

The Debug Log window displays warnings and errors that occur during 2108 Development Environment operations. This includes problems encountered when loading or saving 2108 Project files. The Debug Log window is activated via the Options Menu or toolbar. Function key F3 toggles between the Debug Log and the 2108DevEnv window. Scroll bars and key commands (PgUp, PgDn, Home, etc.) are used to move around the window. Push buttons are provided to copy the contents to the Windows clipboard or to clear the display.

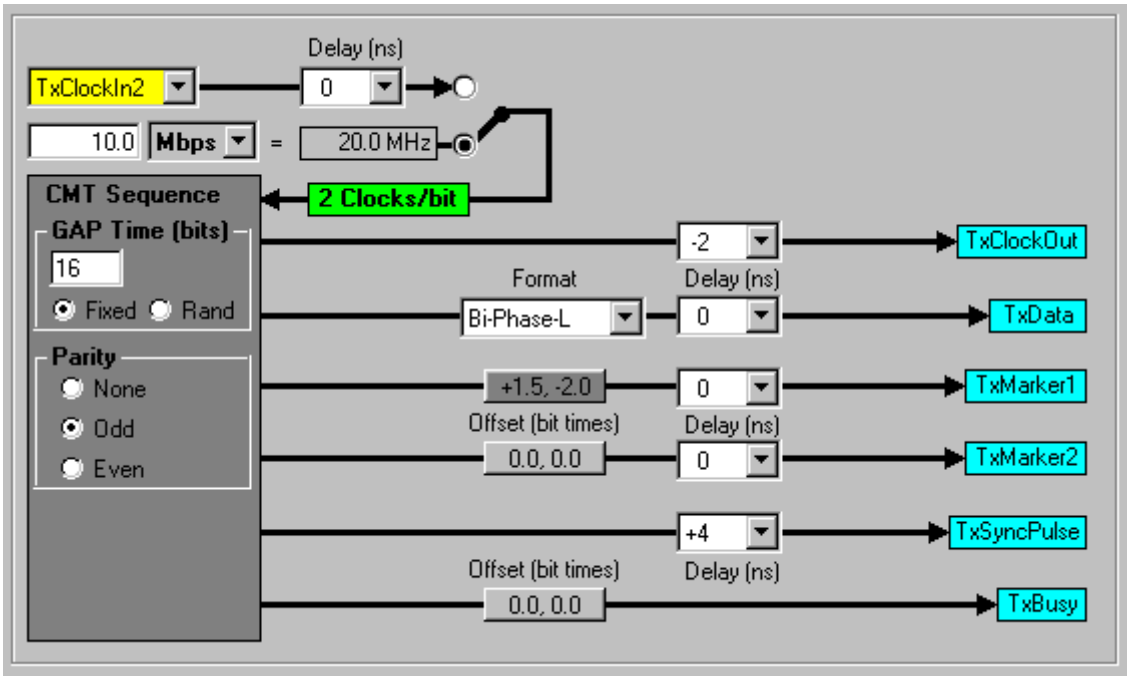



All of the information displayed in this window is also written to a log file (2108DevEnv.log) located in the 2108 Development Environment's home directory.

# 3. Transmit Channel

## 3.1 Transmitter

The Transmitter window defines timing and formatting for a transmit channel's serial output signal, TxData. The transmit clock is selected from an internal source and two external clock input signals. Other output signals, shown in blue, have timing delays and offsets that are assigned with this window.




The output format defines the encoding (e.g. NRZ-L) that is applied to TxData before it is transmitted. This choice has an affect on many other elements of the transmit channel including clock rate, number of clocks per bit, offset timing and Waveform Parameter Registers. See the following Data Formats section for more information on transmitter encoding.

Some of the data formats require two clocks per bit because a level change is necessary in the middle of the bit (e.g. RTZ). A green box is displayed on the input clock line when this is the case. Note that when this situation occurs, and an external clock is selected, the clock source must be set to a frequency of twice the expected bit rate. You can type a bit rate in the text box and press Enter to see the clock frequency required for the selected format.

An offset button displays a signal's low-to-high and high-to-low offset values. The button is set to a darker gray if either offset is nonzero. Pressing an offset button causes the Offset Dialog to be presented for viewing and changing signal edge placement.

The GAP Time is an inter-message gap value (i.e. TGap) that is shared amongst all Control Memory Tables for this transmit channel. Selecting random (Rand) allows gap times to be randomly assigned from within a specified range. The Parity selection is also a choice that is shared amongst all Control Memory Tables for this channel. Individual tables can override this common parity setting.

## 3.2 Data Formats

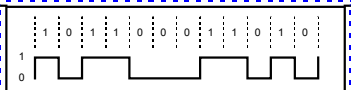
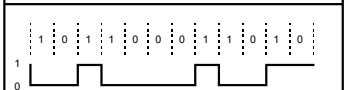
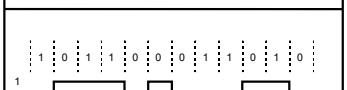
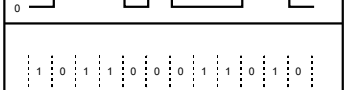
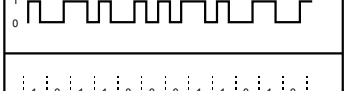
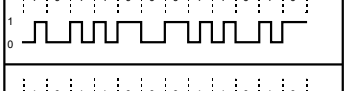
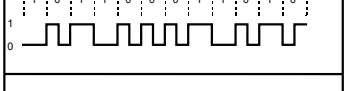
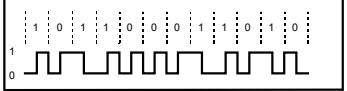
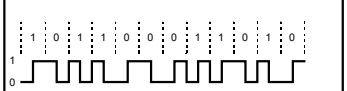
The encoding format of the transmit data has an affect on clock rates and other signal timing. Certain formats have a level change in the middle of a bit period (e.g. Bi-Phase-L). Because of this, the clock rate for those formats must be twice the bit rate in order to properly output the transmit data. This is denoted on the Transmit window by the phrase "**2 Clocks/bit**".

Custom waveforms, defined in the Waveform Parameter Registers window, are also dependent on the format of the transmit data. With a clock that is twice the bit rate, the custom waveforms can include transitions in the middle of a bit period. The minimum size of a waveform is two bits under these conditions.

The offset control on certain output signals is also dependent on the encoding format of the transmit data. Offset is defined for these signals as plus or minus up to three bit times. With a clock that is twice the bit rate, the offsets may be defined in half bit time increments (e.g. +1.5 bit times).

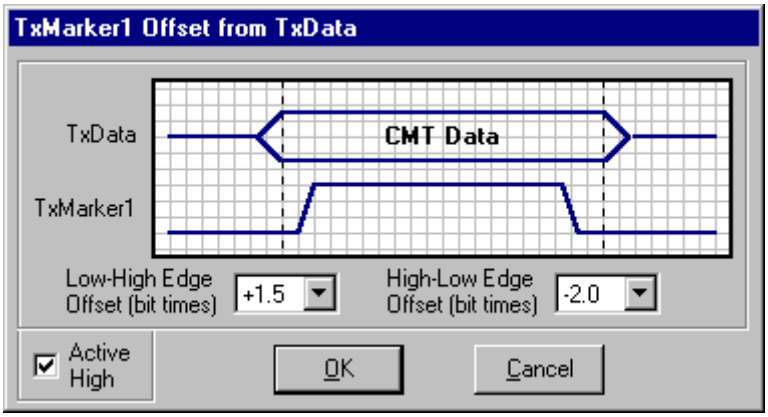
# Model 2108

## Serial Interface Encoding Schemes

| Code Name | Binary Code | Code Definition |
|---|---|---|
| NRZ-L |  | **Non-Return-to-Zero Level**<br>"One" is represented by one level<br>"Zero" is represented by another level lower than the one but not zero |
| NRZ-M |  | **Non-Return-to-Zero Mark**<br>"One" is represented by a change in level<br>"Zero" is represented by no change in level |
| NRZ-S |  | **Non-Return-to-Zero Space**<br>"One" is represented by no change in level<br>"Zero" is represented by a change in level |
| Bi-Phase-L |  | **Bi-Phase Level (Split Phase)**<br>Level change occurs at the center of every bit period<br>"One" is represented by a "One" level with transition to the "Zero" level<br>"Zero" is represented by a "Zero" level with transition to the "One" level |
| Bi-Phase-M |  | **Bi-Phase Mark**<br>Level change occurs at the beginning of every bit period<br>"One" is represented by a midbit level change<br>"Zero" is represented by no midbit level change |
| Bi-Phase-S |  | **Bi-Phase Space**<br>Level change occurs at the beginning of every bit period<br>"One" is represented by no midbit level change<br>"Zero" is represented by a midbit level change |
| DBi-Phase-M |  | **Differential Bi-Phase Mark**<br>Level change occurs at the center of every bit period<br>"One" is represented by no level change at the beginning of the bit period<br>"Zero" is represented by a level change at the beginning of the bit period |
| DBi-Phase-S |  | **Differential Bi-Phase Space**<br>Level change occurs at the center of every bit period<br>"One" is represented by a level change at the beginning of the bit period<br>"Zero" is represented by no level change at the beginning of the bit period |
| AMI |  | **AMI (Alternative Mark Inversion)**<br>"One" is transmitted as a pulse, each successive pulse is of opposite polarity<br>"Zero" is represented by no pulse<br>This type of data transmission results in a reduction of signal loss. |

## 3.3 Offset Dialog

The Offset dialog defines signal edge placement relative to the TxData signal. This dialog is activated from the Transmitter window for the TxMarker1, TxMarker2 and TxBusy signals. The two drop-down lists select the low-high and high-low edge offsets. The waveform display graphically shows the signal edge placement. The offset range is +/-3.0 bit times. ½ bit times are supported when the data format requires two clocks per bit.




The Active High checkbox controls the polarity of the offset signal display. This allows the offsets to be shown according to the programmed, or selected polarity, of the physical signal. TxMarker levels are individually defined within a Control Memory Table window. Output signal polarity is controlled in the Signal Pinout window.

# 3.4 Transmitter Signal Pinout

The Transmitter Signal Pinout window is used to assign output signals to specified pins and to optionally label UUT signals for documentation purposes. The lower left corner of the window has a button that activates a picture of the physical connector. All odd numbered pins on this connector are GND.

| Pin | Connector | Assigned Signal | UUT Signal | Description |
|---|---|---|---|---|
| 2A | TxSig1 | TxData | - | <Output with dynamic reference error> |
| 4A | TxSig2 | TxClock | - | <Output with dynamic reference error> |
| 6A | TxSig3 | TxMarker1 | - | <Static enable output> |
| 8A | TxSig4 | TxMarker2 | - | <Static enable output> |
| 10A | TxSig5 | TxFlagOut1 | EnableA | <Static enable output> |
| 12A | TxSig6 | TxFlagOut2 | EnableB | <Static enable output> |
| 14A | TxSig7 | TxSyncPulse | - | <Static enable output> |
| 16A | TxSig8 | TxBusy | - | <Static enable output> |
| 2B | TxFlagIn1 | | - | <TTL Input flag 1> |
| 4B | TxFlagIn2 | | - | <TTL Input flag 2> |
| 6B | TxClockIn2+ | | - | <Programmable clock positive input> |
| 8B | TxClockIn2- | | - | <Programmable clock negative input> |
| 10B | TxBusy | | - | <TTL Transmitter busy output> |
| 12B | TxSyncPulse | | Scope Trigger | <TTL Transmitter sync pulse> |
| SMA | TxClockIn1 | | - | <High speed ECL input clock> |

Odd numbered pins are GND. TxSig3 through TxSig8 are user-defined from a list of true or complimented signals. TxSig2 is assigned to /TxData or TxClock, depending on the analog driver type. You may enter your own UUT signal names and descriptions.

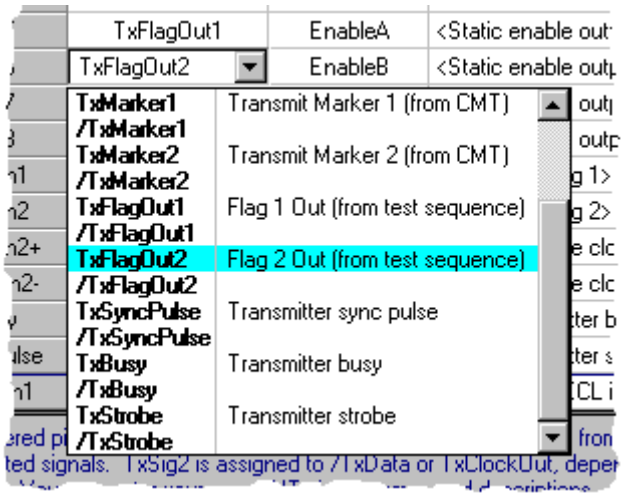
TxSig1 is always assigned to output TxData. TxSig2 outputs either TxClockOut or /TxData, depending on the drive type selected in the Drive Module window. /TxData is selected when the drive type is differential. Outputs TxSig3 through TxSig8 are individually assigned to various transmit signals via a drop-down list.

The UUT Signal and Description columns are provided for user-defined documentation. The angle bracketed text entries (e.g. “<TTL Input flag 2>”) are default descriptions that can be changed. Left click the column to start editing. Press Enter to complete the editing and move to the next row.

For quick reference, the Transmitter Signal Pinout window is easily printed via the toolbar or File Menu.

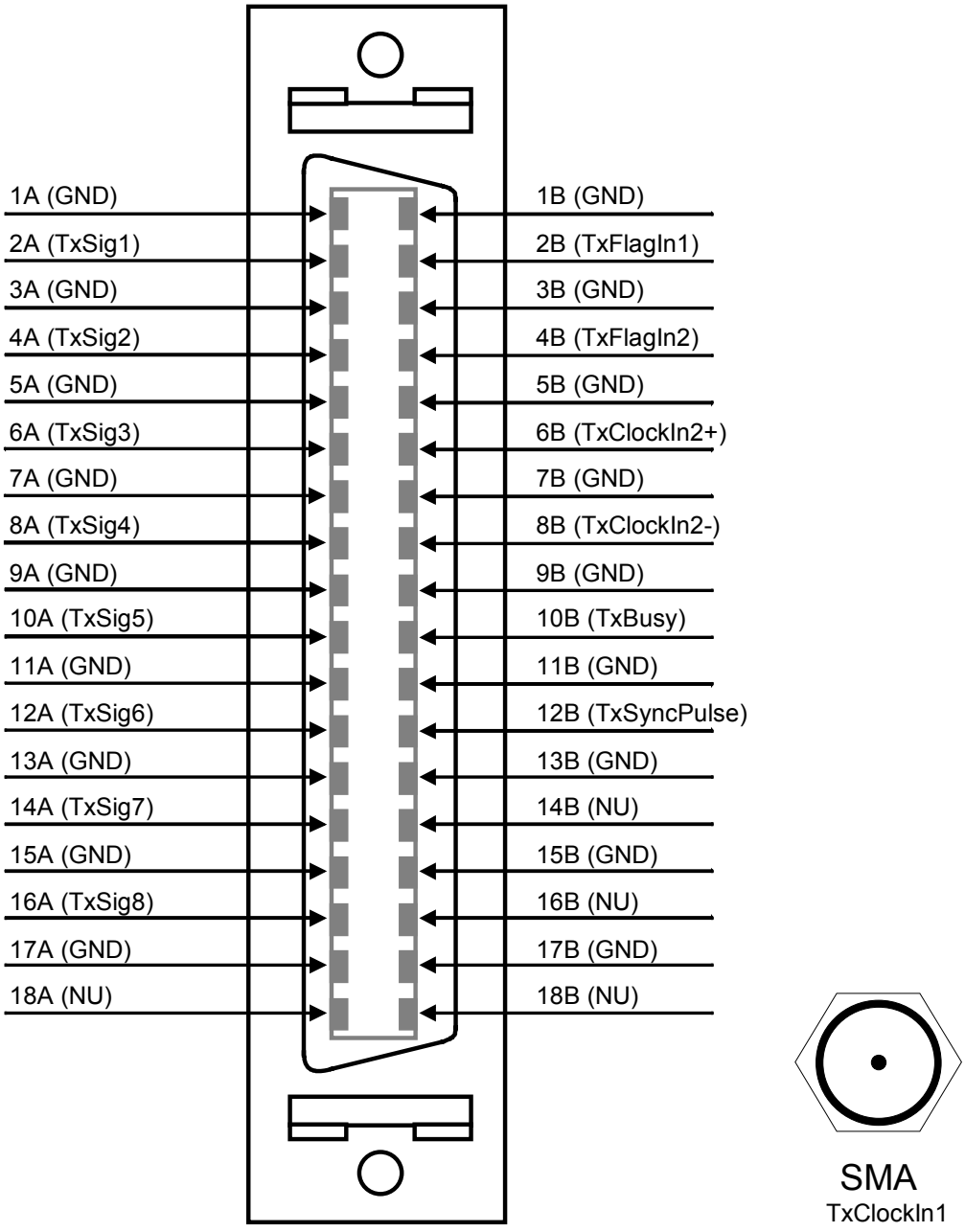
## 3.4.1 Assigning Signals

TxSig3 through TxSig8 are assigned output signals via the Assigned Signal column of the Transmitter Signal Pinout window. Each output can be disabled or assigned to a selection of transmit signals and their complements.



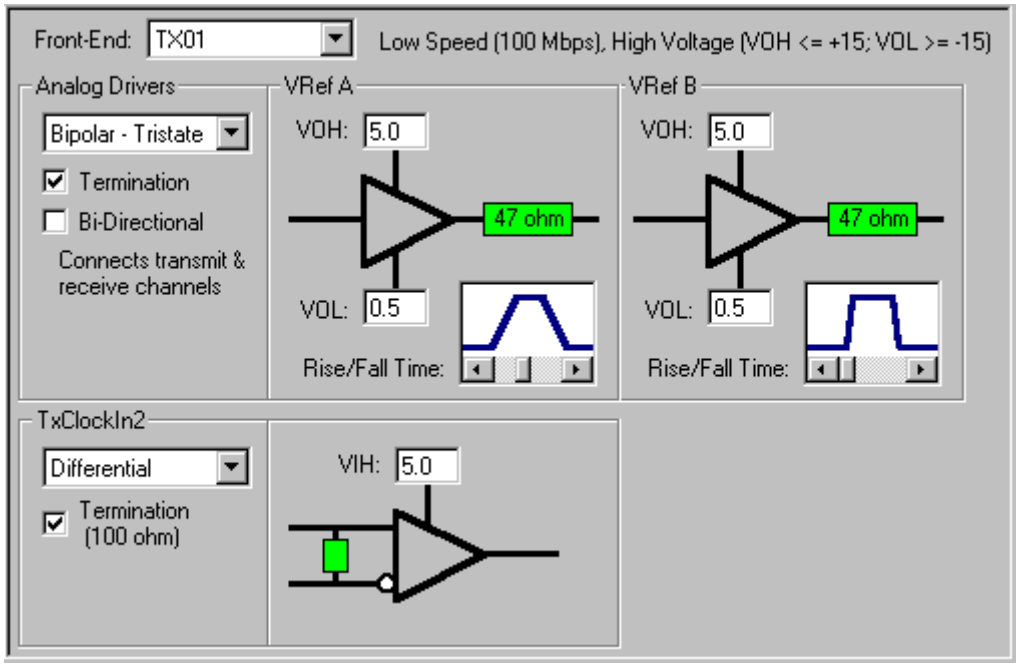
## 3.4.2 Transmit Connector

The Transmit Connector dialog is displayed when a button on the lower left corner of the Transmitter Signal Pinout window is pressed. Two buttons, in the upper right corner of the dialog, allow it to be closed or printed. The dialog can also be moved and resized via the standard Windows mechanisms.



# 3.5 Drive Module

The Drive Module window defines the signal characteristics of the transmitted signals. It also defines the input characteristics of TxClockIn2. VRef A and VRef B are used when defining Waveform Parameter Registers and when a voltage excursion error is instigated with the Execution Manager. VRef A defines the voltage levels for TxData.




The installed Front-End module for a transmit channel controls the output voltage range and clock rate. A module should be selected from the drop-down list that matches the hardware configuration being defined.

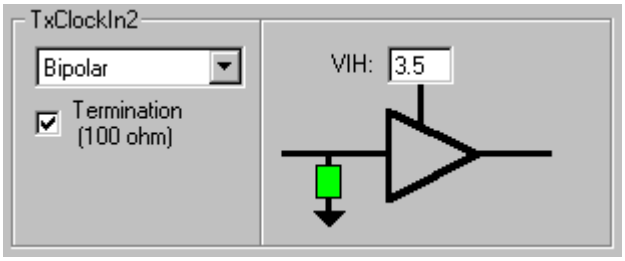
Termination can be added on the output or input signals by a left mouse click on the appropriate Termination checkbox. Output termination is a serial impedance of 47 ohms. Input termination is a parallel impedance of 100 ohms.

The Bi-Directional option allows a transmit channel and a companion receive channel to operate a single pin as both input and output.

Output Rise/Fall time is set by adjusting the slide control under the small waveform display.

## 3.5.1 TxClockIn2

Two different external transmit clocks are available on the Transmitter window. TxClockIn1 is a high speed ECL input that resides on a separate SMA connector. TxClockIn2 has a programmable threshold and interface that is controlled in the Drive Module window. A drop-down list selects between a bipolar or differential interface. A high threshold voltage is specified here as well as the selection of an optional input impedance of 100 ohms.



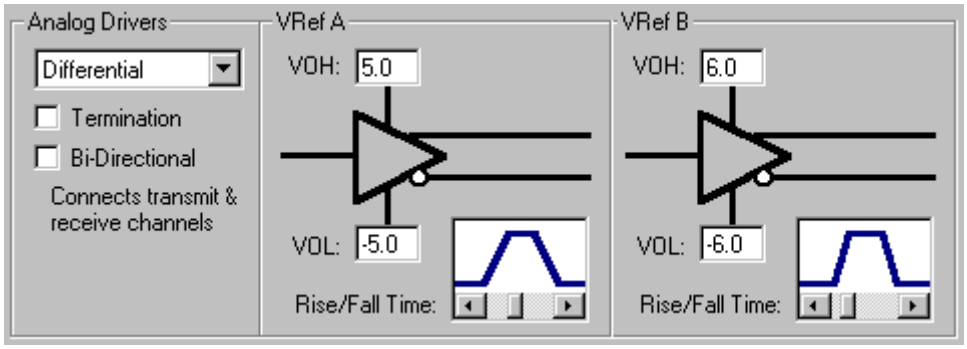

The threshold voltage range is dependent on the transmitter front-end module. The voltage ranges for the front-end modules are shown below.

| Front-End | Bipolar | Differential |
|---|---|---|
| **TX01** | +/-15V | +/-15V |
| **TX02** | +8/-3V | +/-5V |

## 3.5.2 Analog Drivers

### 3.5.2.1 Differential

This selection, within the Drive Module window, causes TxData to be output as a differential signal. High and low output levels (VOH and VOL) are defined for the differential signal. Selecting termination adds 47 ohm series resistors on each of the outputs. Selecting a differential output causes TxData and /TxData to be connected to the top two outputs of the Transmitter Signal Pinout.

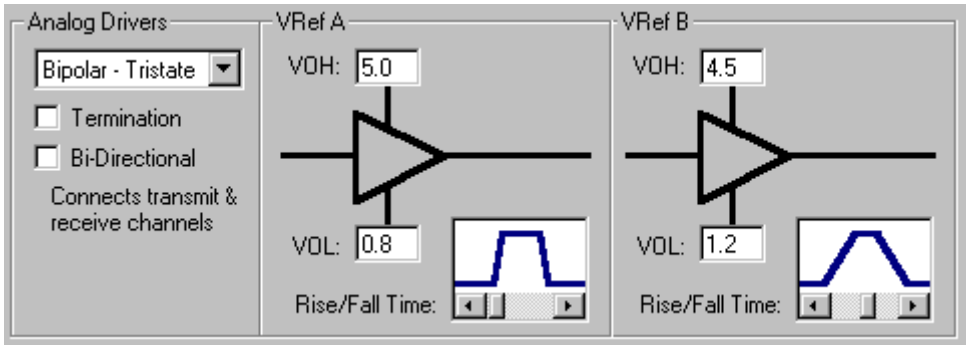



The output voltage ranges are dependent on the transmitter front-end module. In all cases the high output voltage must be greater than the low output voltage (i.e. VOH > VOL). The voltage ranges for the front-end modules are shown below.

| Front-End | Bipolar | Differential |
|---|---|---|
| **TX01** | +/-15V | +/-15V |
| **TX02** | +8/-3V | +8/-3V |

## 3.5.2.2 Bipolar - Tristate

This selection, within the Drive Module window, causes TxData to be output as a bipolar, tristate signal. High and low output levels (VOH and VOL) are defined for the bipolar signal. Selecting termination adds a 47 ohm series resistor on the output. Selecting a bipolar output causes TxData and TxClockOut to be connected to the top two outputs of the Transmitter Signal Pinout.

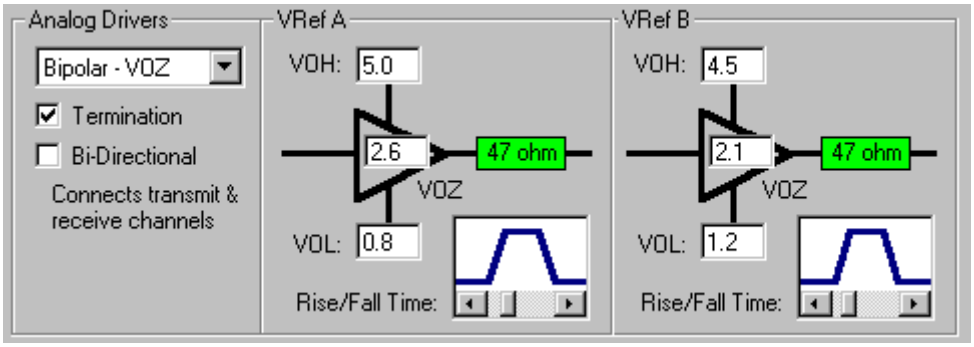



The output voltage ranges are dependent on the transmitter front-end module. In all cases the high output voltage must be greater than the low output voltage (i.e. VOH > VOL). The voltage ranges for the front-end modules are shown below.

| Front-End | Bipolar | Differential |
|---|---|---|
| **TX01** | +/-15V | +/-15V |
| **TX02** | +8/-3V | +8/-3V |

## 3.5.2.3 Bipolar - VOZ

This selection, within the Drive Module window, causes TxData to be output as a bipolar, 3-state signal. High, low and mid-range output levels (VOH, VOL and VOZ) are defined for the bipolar signal. Selecting termination adds a 47 ohm series resistor on the output. Selecting a bipolar output causes TxData and TxClockOut to be connected to the top two outputs of the Transmitter Signal Pinout.

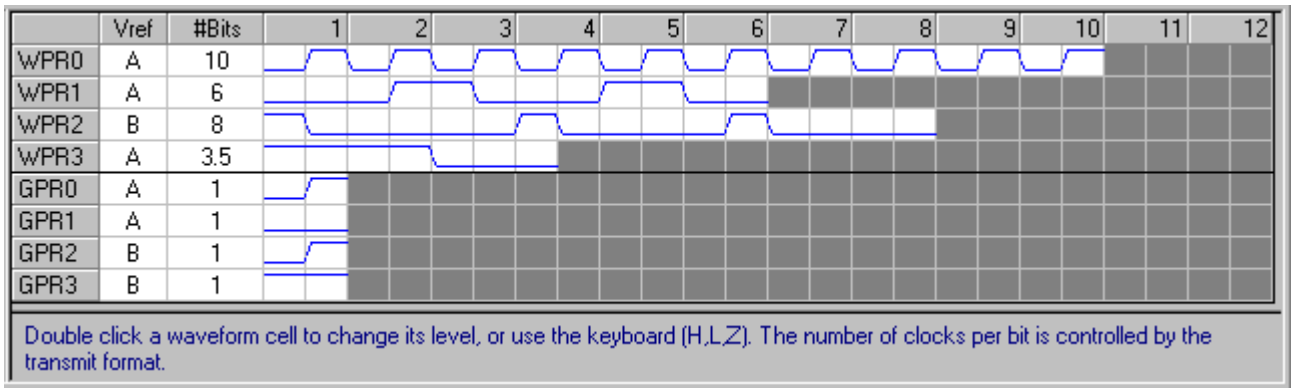



The output voltage ranges are dependent on the transmitter front-end module. In all cases the high output voltage must be greater than the low output voltage (i.e. VOH > VOL). The voltage ranges for the front-end modules are shown below.

| Front-End | Bipolar | Differential |
|---|---|---|
| **TX01** | +/-15V | +/-15V |
| **TX02** | +8/-3V | +8/-3V |

# 3.6 Waveform Parameter Registers

The Waveform Parameter Registers (WPR) window is used to define custom signal waveforms for including in a Control Memory Table. Custom waveforms are often required for packet synchronization or for inter-message gaps between packets.




The Vref column assigns a voltage reference to the waveform. The two voltage references (A, B) are defined in the Drive Module window.

The #Bits column controls the size of the waveform. The maximum length is 12 bits and the minimum is either two or four bits. The waveform length selections, and number of cells per bit, are based on the transmitter data format specified in the Transmitter window.

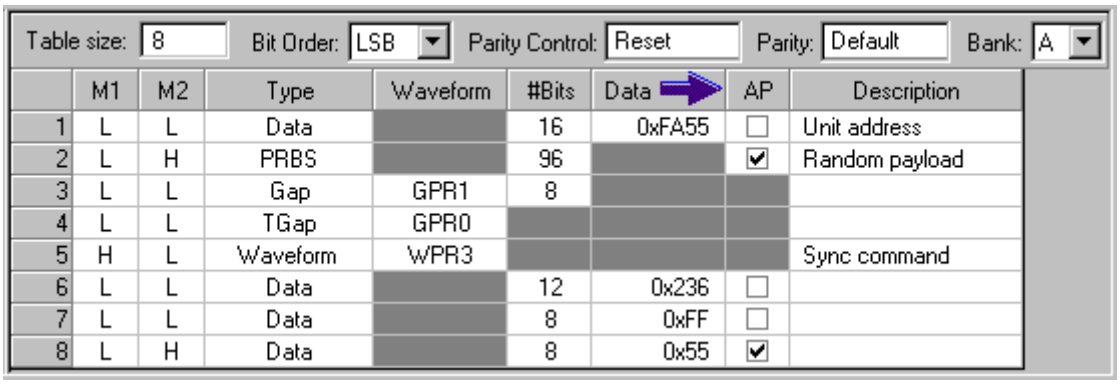
To define a waveform pattern, double left mouse click a waveform cell to change its level. Or you can use the keyboard to enter a level (e.g. H, L, Z) and advance to the next cell.

The Gap waveforms (GPR0-3) are typically used when an inter-message gap is defined in a CMT. Gap waveforms are restricted to a length of one bit. Depending on the data format, this results in one or two cells that can be defined.

# 3.7 Control Memory Tables

A Control Memory Table (CMT) window defines one or more entries that are formatted to generate a serial data output. The entries are output in ascending order. Each entry has a set of characteristics that control its format, size and content. Along the top of the CMT window are controls for table size, bit order, parity and memory bank selection.

Table size (i.e. number of entries) is entered by a user in the designated text box. Pressing the Enter key causes the new size to be displayed. The displayed size is automatically updated to reflect Edit Menu operations. The Bit Order selection is also shown by an arrow at the top of the Data column. See the following Parity section for more information on the parity settings for a CMT. The Memory Bank for storing the CMT is selected from A or B.




| | M1 | M2 | Type | Waveform | #Bits | Data | AP | Description |
|---|---|---|---|---|---|---|---|---|
| 1 | L | L | Data | | 16 | 0xFA55 | ☐ | Unit address |
| 2 | L | H | PRBS | | 96 | | ☑ | Random payload |
| 3 | L | L | Gap | GPR1 | 8 | | | |
| 4 | L | L | TGap | GPR0 | | | | |
| 5 | H | L | Waveform | WPR3 | | | | Sync command |
| 6 | L | L | Data | | 12 | 0x236 | ☐ | |
| 7 | L | L | Data | | 8 | 0xFF | ☐ | |
| 8 | L | H | Data | | 8 | 0x55 | ☑ | |

Individual and multiple entries can be cut, copied, pasted and deleted via the toolbar or Edit Menu. Copied entries can be pasted in other CMTs. Select a single entry with a left click in the leftmost column. Select multiple rows by drag-selecting in the leftmost column. Right click to display a pop-up menu of editing commands. The Data and Description columns can be individually selected and copied or pasted.

Marker signals (M1 = TxMarker1, M2 = TxMarker2) are set high or low for each entry in a CMT. Their edges are delayed or offset via the Transmitter window. The Type column defines the content of the entry's serial bit stream. CMT types include user-defined data, randomly generated data and several forms of user-defined waveforms.

The Waveform column selects between custom waveforms that have been defined by the Waveform Parameter Registers window. See the following Waveform section for more information on selecting waveforms.

The #Bits column defines the size (i.e. number of bits) of the data or gap waveform to generate. Data entries are restricted from 4 to 48 bits. PRBS and Gap types are allowed a range of 4 to 64K bits.

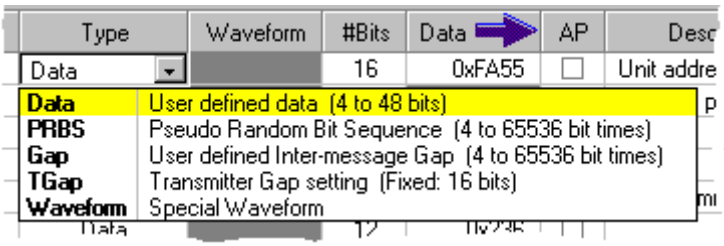
The Data column is used for entries that have been designated as type Data. Left click the column to start editing. Press Enter to complete the editing and move to the next row. Data entries are limited to the bit size specified. Use a leading '0x' or a trailing 'b' to enter hexadecimal or binary data, respectively. Values in the Data column are displayed based on the data format selected via the toolbar or Options Menu.

The Append Parity (AP) column controls whether a parity bit is to be attached to the end of the data stream for that entry. This is valid for Data and PRBS types. The AP column is hidden if Parity Control is set to Disable. The Description column is optionally used to label or document entries in the CMT. This can be helpful when identifying unique sections of a header or data packet. Press Enter to complete a description entry and move to the next row.

Two default CMTs, Idle and Standby, are always present in a 2108 Transmit Channel. The Idle CMT controls transmitter output when no other CMT is active. The Standby CMT sets the conditions for when the transmit drivers are enabled. The Standby CMT is restricted to a single entry.

## 3.7.1 CMT Type

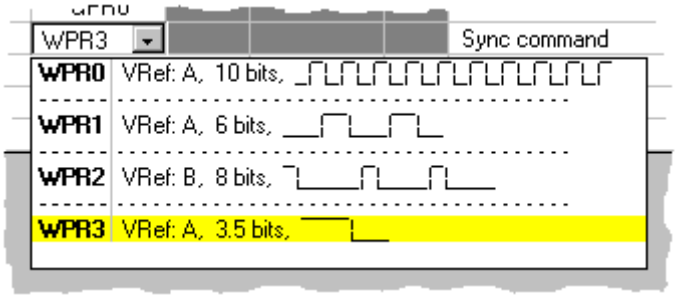
Each entry, in a Control Memory Table window, is defined as a particular type. The selected type determines the set of parameters available for generating a serial data output. Once an entry's type is selected, unnecessary parameter fields are disabled (i.e. grayed-out).




A Data type outputs the specified value, for the specified number of bits, in the specified direction (MSB or LSB). A Pseudo-Random (PRBS) type outputs a random bit stream for the specified number of bits (4-65K). The two Gap types output a custom gap waveform for the specified number of bits. TGap uses a bit size defined on the Transmitter window (and shown in the drop-down list). Waveform outputs one of the custom waveforms specified on the Waveform Parameter Registers window and selected via the Waveform column.

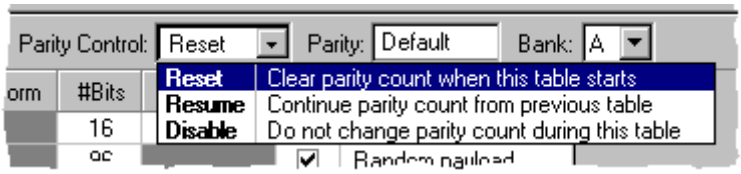
## 3.7.2 Waveform

The Waveform column, on the Control Memory Table window, selects among the custom waveforms defined by the Waveform Parameter Registers window. The selection list contains GPR0-GRP3 for Gap entry types and WPR0-WPR3 for Waveform entry types. Note that the drop-down list includes the voltage reference, size and graphical representation for each of the waveforms.

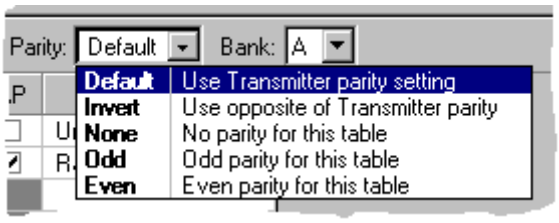



## 3.7.3 Parity

CMT parity options are available along the top of the Control Memory Table window. Parity Control is used to reset, resume or disable parity for the corresponding table. Resume allows parity to be accumulated during the execution of several tables. Reset clears parity at the beginning of this table. Disable preserves the current parity value and does not modify it during this table's execution.




The type of parity to accumulate is controlled by the Parity combo list. Default and Invert use the global parity setting that is defined on the Transmitter window. Using a global parity setting makes it easier to change parity for all CMTs at once. Alternately, the CMT's parity generation can be independently set to None, Odd or Even.

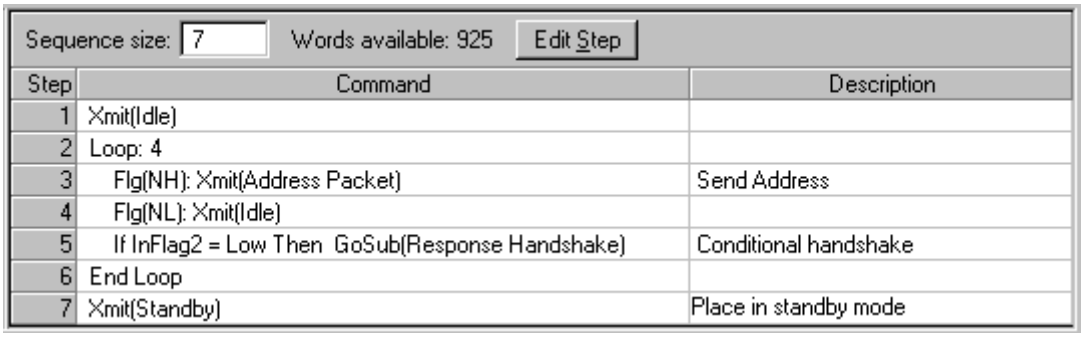



The Add Parity (AP) column has a checkbox for identifying when a parity bit should be added to the serial output signal. The AP column is hidden if parity is disabled for this CMT.

# 3.8 Test Sequences & Test Subroutines

A Test Sequence window defines one or more steps that coordinate Control Memory Table execution, looping and conditional branching. Each step has a set of parameters that control its output, type and behavior. Along the top of the Test Sequence window are controls for sequence size, available words, step editing and subroutine activation.

Sequence size (i.e. number of steps) is entered by a user in the designated text box. Pressing the Enter key causes the new size to be displayed. The displayed size is automatically updated to reflect Edit Menu operations. The **Words available** label identifies the approximate number of memory locations that remain for test sequences. The displayed value is an approximation because some step types require multiple memory locations.




The Edit Step button activates a Test Sequence Step dialog for editing the topmost selected step. Alternately, double left click a sequence step to begin editing it. The Subroutine checkbox indicates when a sequence is to be a subroutine. Only subroutine sequences can be called through the RxTrigger Subroutine Table or by a GoSub command from another sequence.

Individual and multiple steps can be cut, copied, pasted and deleted via the toolbar or Edit Menu. Copied steps can be pasted in other sequences. Select a single step with a left click in the leftmost column. Select multiple steps by drag-selecting in the leftmost column. Right click to display a pop-up menu of editing commands. The Description column can be individually selected and copied or pasted.

The test sequence steps are displayed in a pseudo-code text format. This makes it easy to understand the event ordering and conditional behavior. Indentation is used to identify matched loop commands. Unmatched loops are highlighted by a red background. The Description column is optionally used to label or document test sequence steps. This can be helpful when identifying unique output behavior or conditional responses. Press Enter to complete a description entry and move to the next row.

There are four categories of test sequence steps. The following briefly describes each of them. Subsequent sections provide a more detailed description of their options and behavior.

An Output step controls CMT execution, TxFlagOut levels and RxStart. Output steps may be conditionally executed or delayed until a specified condition is achieved.

A Jump step moves the execution flow to an internal step or to another test sequence. External jump destinations cannot be subroutines. Jump commands are either unconditional or based on a selection of external conditions and states.

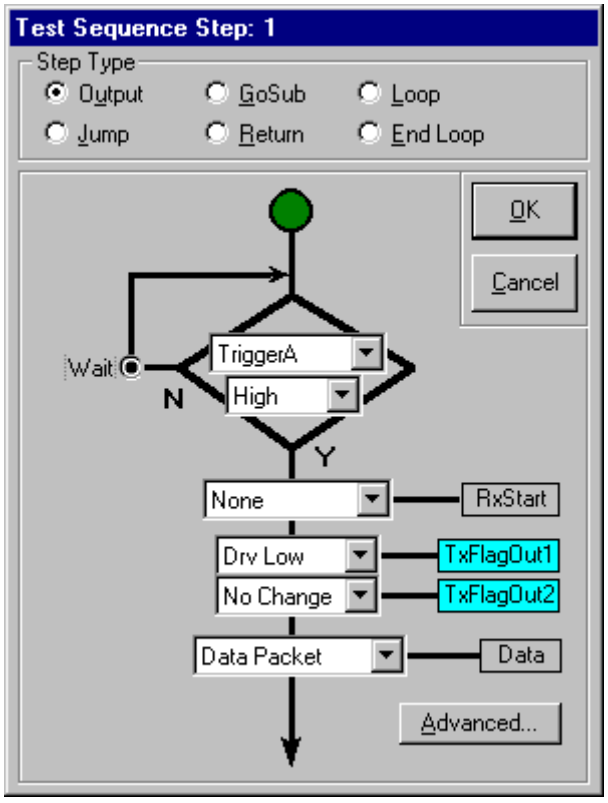
A GoSub step temporarily redirects the execution flow to another test sequence. All GoSub destinations must be designated as subroutines. GoSub commands are either unconditional or based on a selection of external conditions and states. A companion Return command restores execution to the calling sequence (only available in Subroutines).

A Loop step creates a looping construct in the test sequence. This allows a group of steps to be repeated for a specified number of times. A companion End Loop command identifies the close of a loop.

## 3.8.1 Edit Step

### 3.8.1.1 Output

An Output test sequence step manages several different output options and conditions. The output options include executing a CMT, changing TxFlagOut signal levels and initiating a record sequence. The output actions can be conditionally executed based on a selection of flags (e.g. TxFlagIn1, RxTrigger, etc.) and states (e.g. High, Inactive, etc.). Optionally, the step can wait until the designated condition is met.

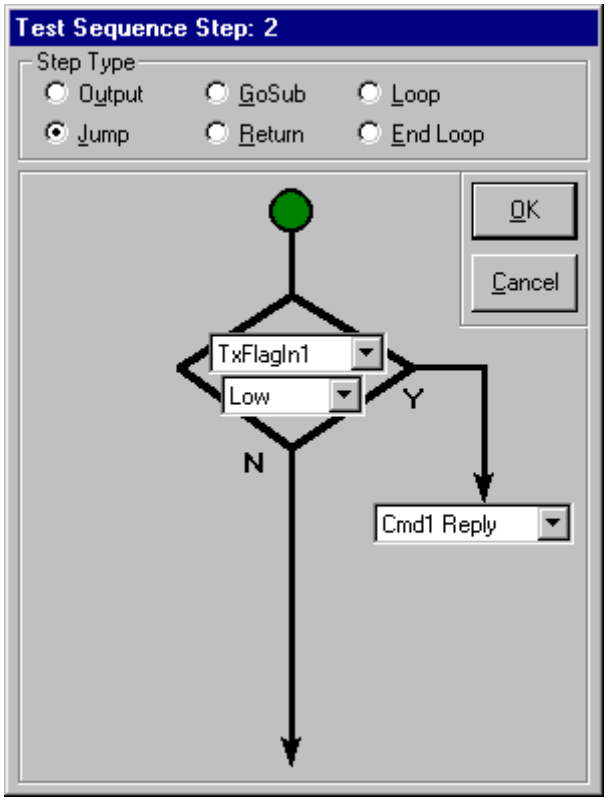



The RxStart drop-down list identifies a record sequence step to activate. Record sequences are armed for a single step or a series of steps. The levels of TxFlagOut1 and TxFlagOut2 may be set to high or low, or left unchanged, during an Output test sequence step.

If a CMT is selected for execution, the Advanced button is displayed. Pressing the Advanced button activates the Advanced CMT Control dialog. This dialog presents additional control options for CMT execution. If advanced options are selected, the button font is set to **bold**.

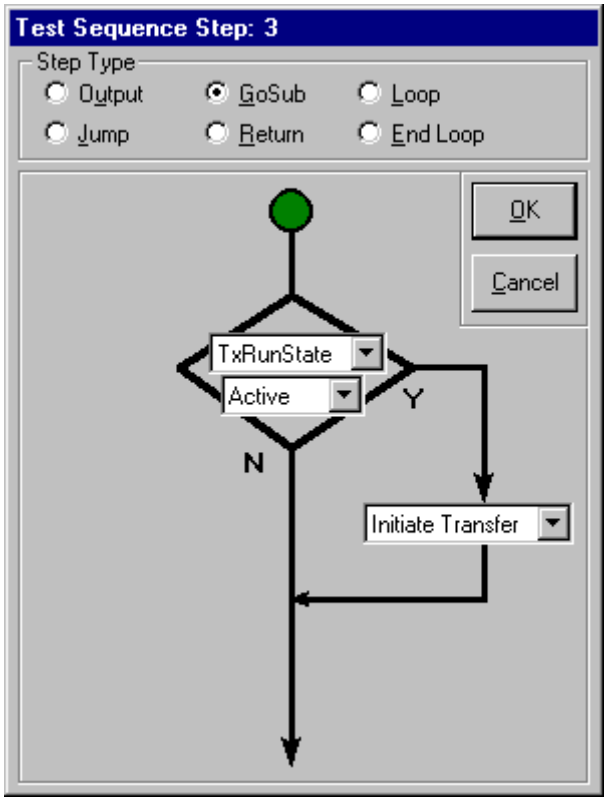
## 3.8.1.2 Jump

A Jump step changes the execution flow of a test sequence. The destination for a jump is either a step within the current sequence (e.g. <Step 4>), or the first step of a different test sequence. You cannot jump to a test sequence that has been designated as a subroutine. A jump can be conditionally executed based on a selected flag (e.g. TxFlagIn2, TriggerA, etc.) and state (e.g. High, Inactive, etc.).



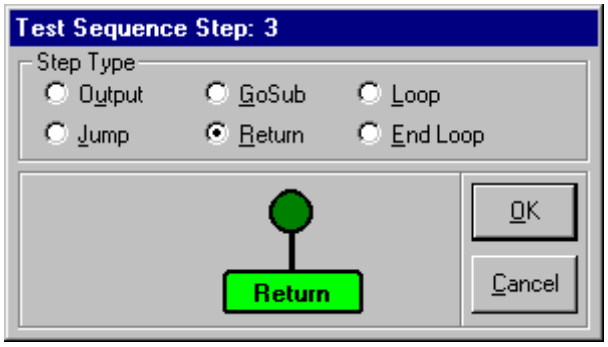
## 3.8.1.3 GoSub & Return

A GoSub command passes execution to a test sequence subroutine. Once the subroutine is complete, the execution flow resumes at the following step. The lines and arrows on the dialog denote the possible execution paths. A GoSub can be conditionally executed based on a selected flag (e.g. TxRunState, TriggerA, etc.) and state (e.g. High, Inactive, etc.).



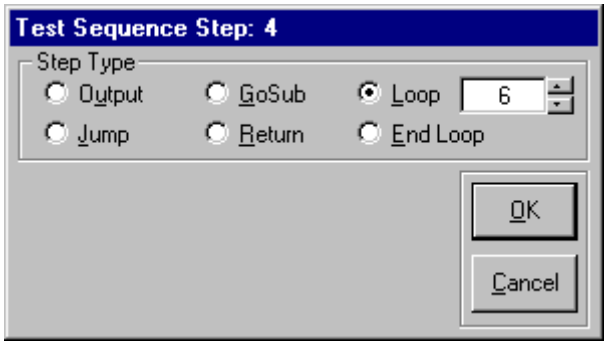

The RxTrigger condition provides a way to dynamically respond to a receive trigger that occurs. When the RxTrigger condition is set to True, the destination drop-down list includes an entry for a subroutine vector table (i.e. <Subroutine Table>). This refers to the RxTrigger Subroutine Table that associates subroutines with triggers defined in a Record Sequence window. Thus, the executed subroutine is dependent on the trigger that causes RxTrigger to be True.

The Return command is used to exit a subroutine sequence and resume execution in the calling sequence. A test sequence designated as a subroutine is automatically given a Return command following its last step. An explicit Return can be used to exit prior to the end of a sequence. This is useful when a subroutine is waiting for a particular condition, or needs to be terminated early.



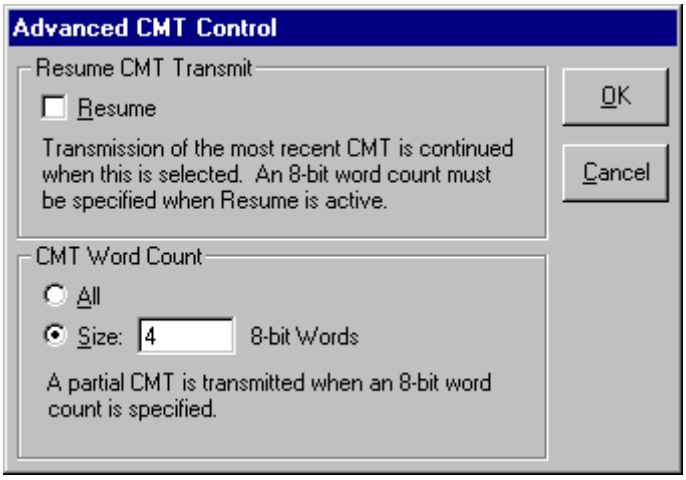
## 3.8.1.4 Loop & End Loop

These commands define the beginning and end of a loop construct. Loop and End Loop entries must be matched or a red highlight is displayed in the Test Sequence window. Loops can be repeated for up to 64K. The loop count can be incremented and decremented with the up/down buttons on the right side of the text box. Alternately, the loop count can be entered via the keyboard. A loop may be exited through the use of a Jump or Return command.



# 3.8.1.5 Advanced CMT Control

The Advanced CMT Control dialog is activated from an Output step type on the Edit Sequence window. This is used for discreet control over the CMT output. The Resume option is used to restart a CMT that has been partially output. This allows a CMT's execution to be paced by external conditions. Note that the Resume option requires that a Word Count also be specified. The Word Count is for 8-bit words. A 16-bit CMT entry would correspond to two 8-bit words. To run the whole CMT, simply select All.

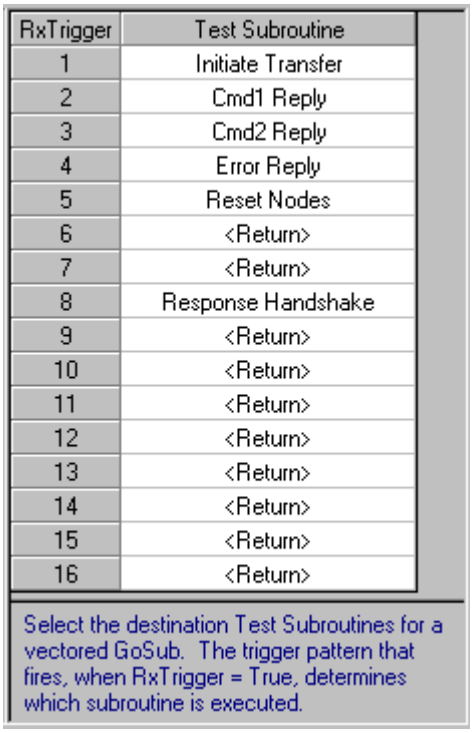


# 3.9 RxTrigger Subroutine Table

The RxTrigger Subroutine Table window is used to define indirect subroutine calls based on received triggers. This window is only necessary for test sequence subroutine calls where **RxTrigger = True** and the destination is **<Subroutine Table>**. If those two conditions are not present, this table can safely be ignored.

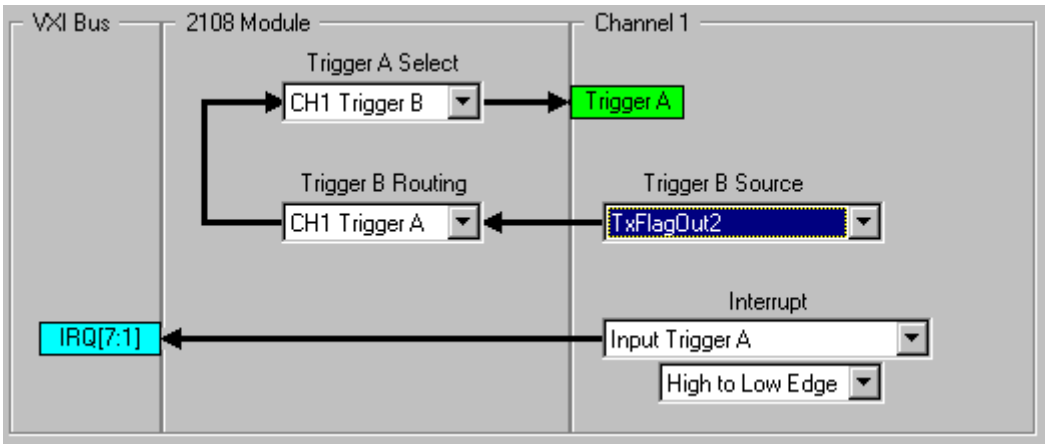
This is essentially an indirect jump, or vector, table that is activated by the occurrence of a received trigger. Each entry in the table corresponds to a possible record trigger defined in the Record Sequences window. The table is used to complete a test sequence GoSub command where the condition is: RxTrigger = True, and the destination is: <Subroutine Table>. The subroutine executed is determined by the trigger that occurs.

Left click the Test Sequence column to activate a drop-down list of the subroutines defined for the channel. Select one to define a destination for a trigger-based indirect subroutine call. Select <Return> to have execution return immediately to the calling sequence.



# 3.10 Transmitter VXI Triggers & Interrupts

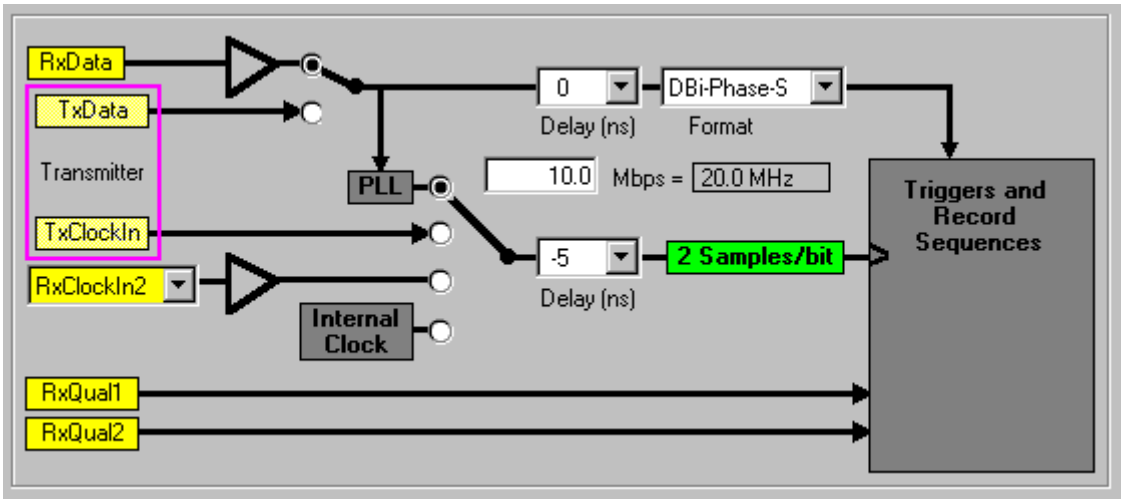
The Transmitter VXI Triggers & Interrupts window controls the source for Trigger A and Trigger B. Trigger B may be routed to VXI triggers (4-7) or to Trigger A of other channels within the 2108 module. This window also selects a signal for the VXI interrupt bus.



# 4. Receive Channel

## 4.1 Receiver

The Receiver window defines timing and formatting for a receive channel's serial input signal. The receive data is selected from an external input or an internal link to a companion transmit channel. The receive clock is derived from the receive data, or selected from a companion transmit channel, two external clock inputs or an internal clock. Both receive data and clock have timing delays that are assigned with this window.




The input format defines the encoding (e.g. NRZ-L) that is expected on the receive data. This choice has an affect on many other elements of the receive channel including clock rate, number of clocks per bit, trigger patterns and trigger count. See the following Data Formats section for more information on data encoding issues.

Some of the data formats require two clocks per bit because a level change is necessary in the middle of the bit (e.g. RTZ). A green box is displayed on the input clock line when this is the case. Note that when this situation occurs, and an internal clock is selected, the clock source must be set to a frequency of twice the expected bit rate. You can type a bit rate in the text box and press Enter to see the clock frequency required for the selected format.

## 4.2 Data Formats

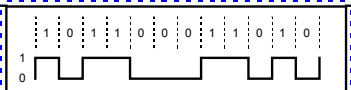
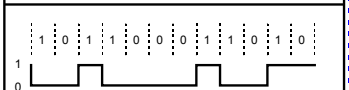
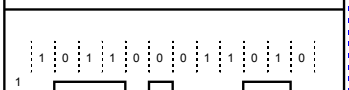
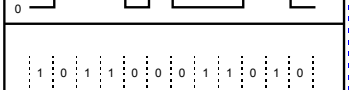
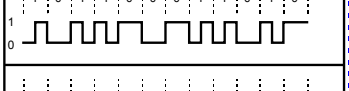
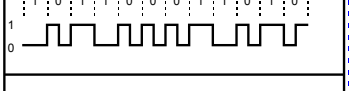
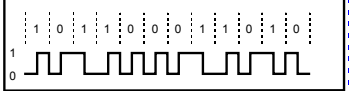
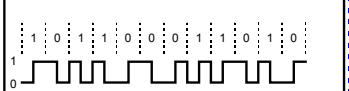
The encoding format of the receive data has an affect on clock rates and trigger patterns. Certain formats have a level change in the middle of a bit period (e.g. Bi-Phase-L). Because of this, the clock rate for those formats must be twice the bit rate in order to properly sample the receive data. This is denoted on the Receiver window by the phrase "**2 Samples/bit**".

Trigger patterns, defined in the Record Sequences window, are also dependent on the format of the receive data. If the pattern is defined for undecoded ("raw") signal levels (H,L,Z,X), then it must be based on the expected type of encoding. One way to think of the encoded data is to consider how it would look if the signal was measured and displayed on an oscilloscope.

Some serial encoding schemes are only based on level changes rather than specific levels or transitions (i.e. low to high). For these formats, triggers are harder to define because the direction of a level change is dependent on the ending level of the previous bit. Consequently, level change formats (e.g. Bi-Phase-M) require triggers to be defined for two possible starting levels. This is handled automatically by the software, but it does cause the number of triggers to be limited to eight.

# Model 2108

## Serial Interface Encoding Schemes

| Code Name | Binary Code | Code Definition |
|---|---|---|
| NRZ-L |  | **Non-Return-to-Zero Level**<br>"One" is represented by one level<br>"Zero" is represented by another level lower than the one but not zero |
| NRZ-M |  | **Non-Return-to-Zero Mark**<br>"One" is represented by a change in level<br>"Zero" is represented by no change in level |
| NRZ-S |  | **Non-Return-to-Zero Space**<br>"One" is represented by no change in level<br>"Zero" is represented by a change in level |
| Bi-Phase-L |  | **Bi-Phase Level (Split Phase)**<br>Level change occurs at the center of every bit period<br>"One" is represented by a "One" level with transition to the "Zero" level<br>"Zero" is represented by a "Zero" level with transition to the "One" level |
| Bi-Phase-M |  | **Bi-Phase Mark**<br>Level change occurs at the beginning of every bit period<br>"One" is represented by a midbit level change<br>"Zero" is represented by no midbit level change |
| Bi-Phase-S |  | **Bi-Phase Space**<br>Level change occurs at the beginning of every bit period<br>"One" is represented by no midbit level change<br>"Zero" is represented by a midbit level change |
| DBi-Phase-M |  | **Differential Bi-Phase Mark**<br>Level change occurs at the center of every bit period<br>"One" is represented by no level change at the beginning of the bit period<br>"Zero" is represented by a level change at the beginning of the bit period |
| DBi-Phase-S |  | **Differential Bi-Phase Space**<br>Level change occurs at the center of every bit period<br>"One" is represented by a level change at the beginning of the bit period<br>"Zero" is represented by no level change at the beginning of the bit period |
| AMI |  | **AMI (Alternative Mark Inversion)**<br>"One" is transmitted as a pulse, each successive pulse is of opposite polarity<br>"Zero" is represented by no pulse<br>This type of data transmission results in a reduction of signal loss. |

talon INSTRUMENTS

150 E. Arrow Hwy. San Dimas, CA 91773 909-599-0690, Fax 909-599-6529

# 4.3 Receiver Signal Pinouts

The Receiver Signal Pinout window is used to assign output signals to specified pins and to optionally label UUT signals for documentation purposes. The lower left corner of the window has a button that activates a picture of the physical connector. All odd numbered pins on this connector are GND.

| Pin | Connector | Assigned Signal | UUT Signal | Description |
|---|---|---|---|---|
| 2A | RxData+ | | - | <Positive data input> |
| 4A | RxData- | | - | <Negative data input> |
| 6A | RxClockIn2+ | | - | <Positive clock input> |
| 8A | RxClockIn2- | | - | <Negative clock input> |
| 10A | RxQual1+ | | Gate 23 | <Qualifier 1 positive input (Q1)> |
| 12A | RxQual1- | | N/A | <Qualifier 1 negative input> |
| 14A | RxQual2+ | | N/A | <Qualifier 2 positive input (Q2)> |
| 16A | RxQual2- | | N/A | <Qualifier 2 negative input> |
| 18A | RxSig1 | RxArm | Ext Activate | <TTL receiver output 1> |
| 2B | RxTrigValid | | - | <TTL trigger valid tag> |
| 4B | RxTrigNum0 | | - | <TTL trigger number zero> |
| 6B | RxTrigNum1 | | - | <TTL trigger number one> |
| 8B | RxTrigNum2 | | - | <TTL trigger number two> |
| 10B | RxTrigNum3 | | - | <TTL trigger number three> |
| 12B | RxG0Val | | - | <TTL receiver good zero data> |
| 14B | RxG1Val | | - | <TTL receiver good one data> |
| 16B | RxClockOut | | - | <TTL receiver clock> |
| 18B | RxSig2 | RxBusy | Pacing Signal | <TTL receiver output 2> |
| SMA | RxClockIn1 | | - | <High speed ECL input clock> |

Odd numbered pins are GND. RxSig1 and RxSig2 can each be assigned to a different selection of signals. You can enter your own UUT signal names and descriptions.

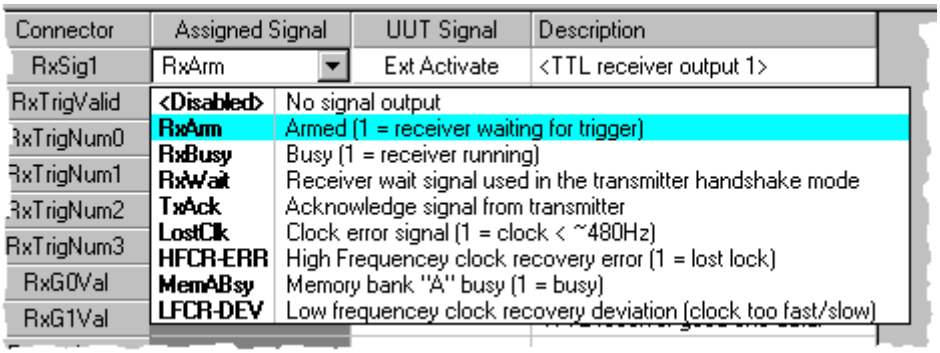
Outputs RxSig1 and RxSig2 are individually assigned to various signals via a drop-down list.

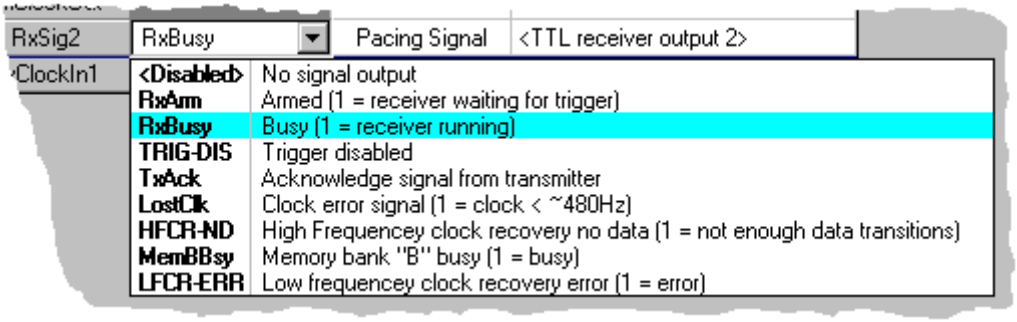
The UUT Signal and Description columns are provided for user-defined documentation. The angle bracketed text entries (e.g. “<Positive data input>”) are default descriptions that can be changed. Left click the column to start editing. Press Enter to complete the editing and move to the next row.

For quick reference, the Receiver Signal Pinout window is easily printed via the toolbar or File Menu.

## 4.3.1 Assigning Signals

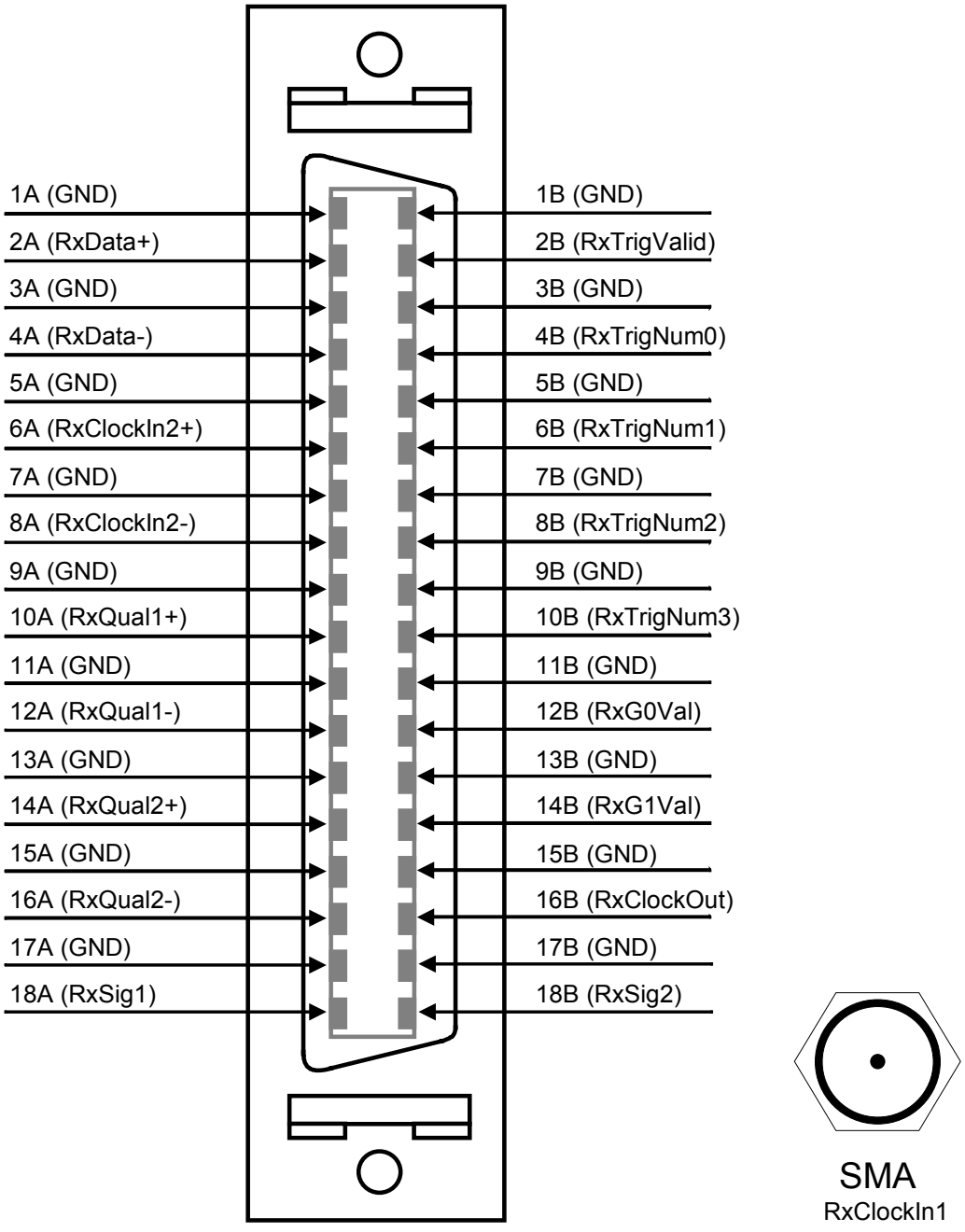
RxSig1 and RxSig2 are assigned output signals via the Assigned Signal column of the Receiver Signal Pinout window. Each output can be disabled or assigned to a different selection of signals and conditions.



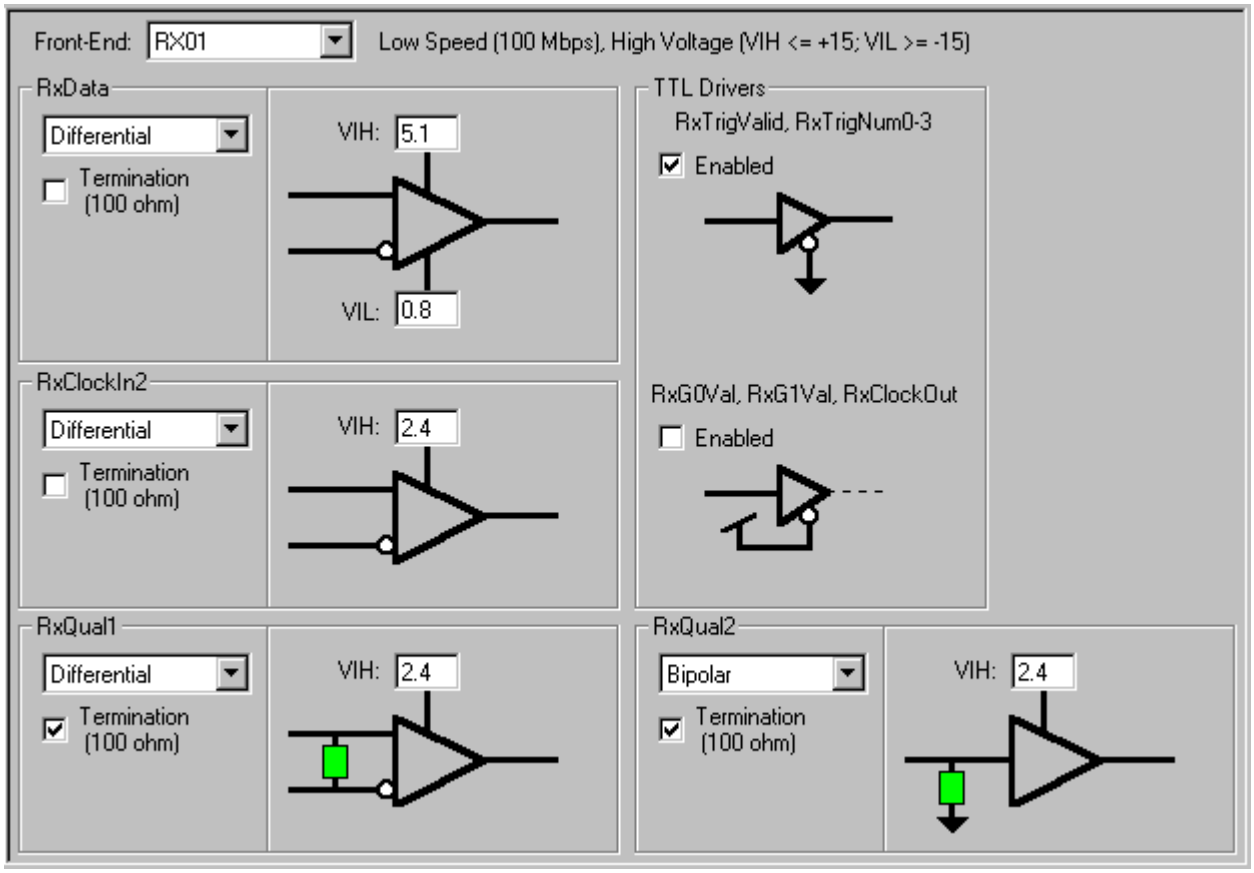
## 4.3.2 Receiver Connector

The Receiver Connector dialog is displayed when a button on the lower left corner of the Receiver Signal Pinout window is pressed. Two buttons, in the upper right corner of the dialog, allow it to be closed or printed. The dialog can also be moved and resized via the standard Windows mechanisms.



# 4.4 Receive Module

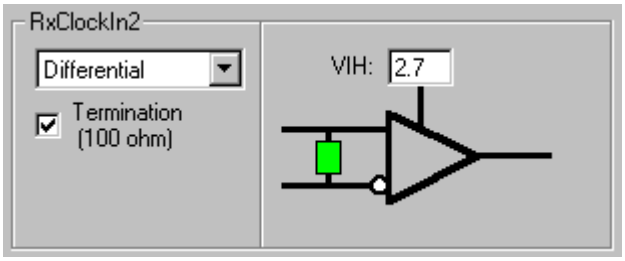
The Receive Module window defines the characteristics of the received data signal. It also defines the input characteristics of RxClockIn2, RxQual1 and RxQual2. Two sets of TTL output signals are enabled, or disabled, via this window. The installed Front-End module for a receive channel controls the input voltage range and clock rate. A module should be selected from the drop-down list that matches the hardware configuration being defined.




Termination can be added on the input signals by a left mouse click on the appropriate Termination checkbox. Input termination is a parallel impedance of 100 ohms. If the TTL output signals are not needed, they can be disabled to minimize signal noise.

## 4.4.1 RxClockIn2

Two different external receive clocks are available on the Receiver window. RxClockIn1 is a high speed ECL input that resides on a separate SMA connector. RxClockIn2 has a programmable threshold and interface that is controlled in the Receive Module window. A drop-down list selects between a bipolar or differential interface. A high threshold voltage is specified here as well as the selection of an optional input impedance of 100 ohms.

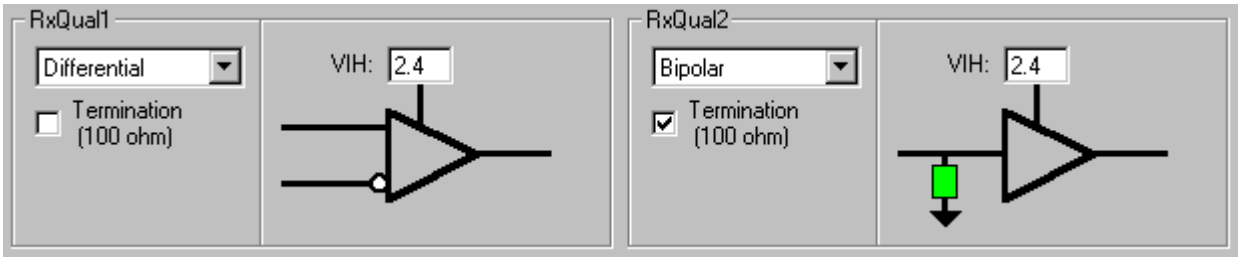



The threshold voltage range is dependent on the receiver front-end module. The voltage ranges for the front-end modules are shown below.

| Front-End | Bipolar | Differential |
|---|---|---|
| **RX01** | +/-15V | +/-15V |
| **RX02** | +8/-3V | +/-5V |

## 4.4.2 Receive Qualifiers

Receive qualifier signals are used to further define triggers in the Record Sequences window (Q1, Q2). These two signals have programmable thresholds and interfaces that are controlled in the Receive Module window. A drop-down list selects between a bipolar or differential interface. A high threshold voltage is specified here as well as the selection of an optional input impedance of 100 ohms.

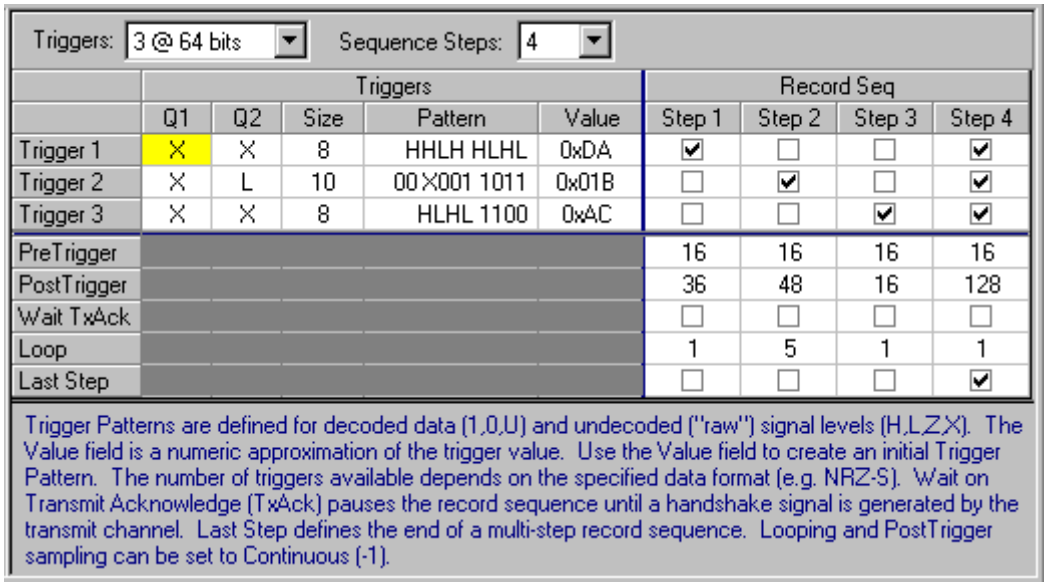



The threshold voltage range is dependent on the receiver front-end module. The voltage ranges for the front-end modules are shown below.

| Front-End | Bipolar | Differential |
|---|---|---|
| **RX01** | +/-15V | +/-15V |
| **RX02** | +8/-3V | +/-5V |

# 4.5 Record Sequences

The Record Sequences window defines the triggers and record sequences that are used to capture serial receive data with a 2108 Receive Channel. This is analogous to setting trigger conditions for a logic analyzer. Along the top of the window are controls for selecting the number of triggers and the number of record sequence steps. As these selections change, the corresponding trigger rows and step columns are added or removed. Up to 16 triggers and 16 record sequence steps may be defined.

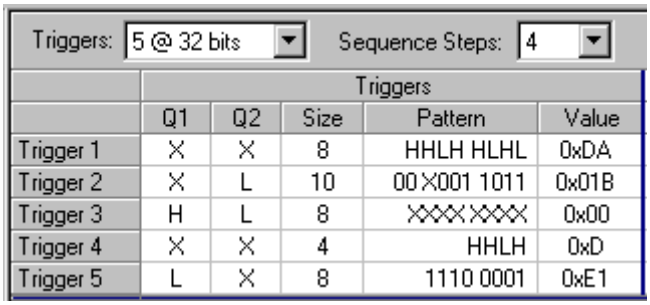



In the Execution Manager, record sequences are armed by step number. Either a single step or a series of steps may be readied for recording. Once armed, they await the occurrence of a trigger condition. One or more triggers can be associated with a record sequence step. If any of the trigger conditions is met, the record sequence step is activated and the incoming serial data is stored.

Often, a succession of events must be detected before recording begins. This is accomplished by creating a series of record sequence steps with triggers that match the necessary events (e.g. Trigger 1 occurs, followed by 2, followed by 3). Each of the preliminary steps would record a minimum number of samples. The final step would record the desired amount of serial data.

## 4.5.1 Record Sequence Triggers

The trigger portion, of the Record Sequences window, defines the size and content of trigger conditions that are used to activate record sequences. Each trigger condition is composed of two qualifier settings and a trigger pattern. The number of triggers, and their maximum width, is specified by the drop-down list at the top of the window (e.g. 4 @ 64 bits). The contents of the drop-down list is dependent on the receive data format.




| | Triggers | | | | |
|---|---|---|---|---|---|
| | Q1 | Q2 | Size | Pattern | Value |
| Trigger 1 | X | X | 8 | HHLH HLHL | 0xDA |
| Trigger 2 | X | L | 10 | 00X001 1011 | 0x01B |
| Trigger 3 | H | L | 8 | XXXX XXXX | 0x00 |
| Trigger 4 | X | X | 4 | HHLH | 0xD |
| Trigger 5 | L | X | 8 | 1110 0001 | 0xE1 |

The receive qualifier signals (Q1, Q2) can be defined to be high, low or don't care (X). These are external inputs with programmable thresholds and interfaces that are set in the Receive Module window. These signals can serve as additional requirements to a record trigger definition, or as the exclusive condition for a trigger (see Trigger 3 in above image).

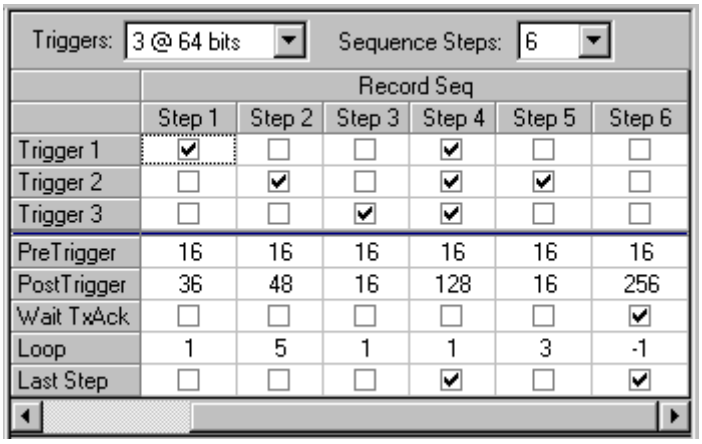
The size column defines the number of bits allotted for a trigger. The minimum size for a trigger is four bits. The maximum size is either 32 or 64 bits, depending on the number of triggers defined and the encoding of the receive data. If only half of the available triggers are used, then they can each be up to 64 bits long.

The trigger pattern is defined by individual bit state characters. The number of bits allowed is controlled by the size field. The bit states supported are for both decoded data (1, 0, U) and encoded signal levels (H, L, Z, X). Note that mixing decoded and encoded bit states in a single pattern is allowed, but it requires a thorough understanding of how the data is ordered and encoded. Trigger patterns can be individually copied and pasted by using editing shortcut keys for copy (**Ctrl+C**) and paste (**Ctrl+V**).

The value column displays a numeric representation of the trigger pattern. Since a trigger pattern can include a mixture of unknown states and encoded data, this value is only an approximation. If a trigger of a particular value is desired, it can be entered in the value field in order to initialize the trigger pattern. The contents of the value column are displayed in the numeric format selected by the Options Menu or toolbar.

## 4.5.2 Record Sequence Steps

The record sequence step portion, of the Record Sequences window, defines the activation events for recording serial receive data. It also controls the amount of data recorded before and after the event occurrence. A step definition includes one or more selected triggers, Pre and Post trigger sample sizes, a repeat count and a last step indicator.




A trigger is selected for a step by a left mouse click in the appropriate checkbox. Triggers are logically Or'd together for a step. In other words, only one of the selected trigger conditions must be met in order for the step to begin recording.

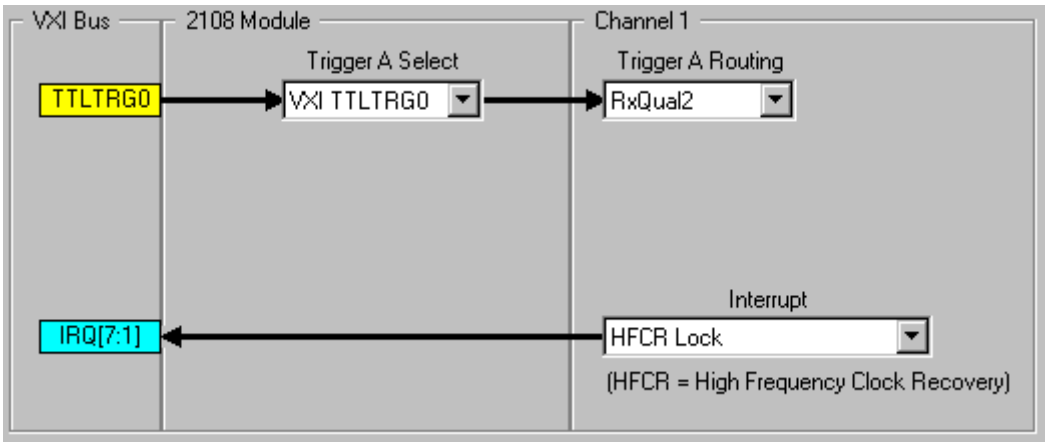
The Pre and Post trigger fields define the number of samples that will be recorded before and after the trigger condition. The Post trigger value can be set to negative one (-1) in order to sample continuously. The Loop field defines the number of times to repeat the step before stopping, or moving to the next step. It also can be set to loop continuously (-1).

The Last Step field is used when a series of record sequence steps are to be applied. The Execution Manager selects a starting step number for a multi-step series. The steps are armed and activated, in numerical order, until the designated 'last step' is completed.

The Wait TxAck option pauses the record sequence until a handshake signal (Transmit Acknowledge) is generated by the transmit channel.

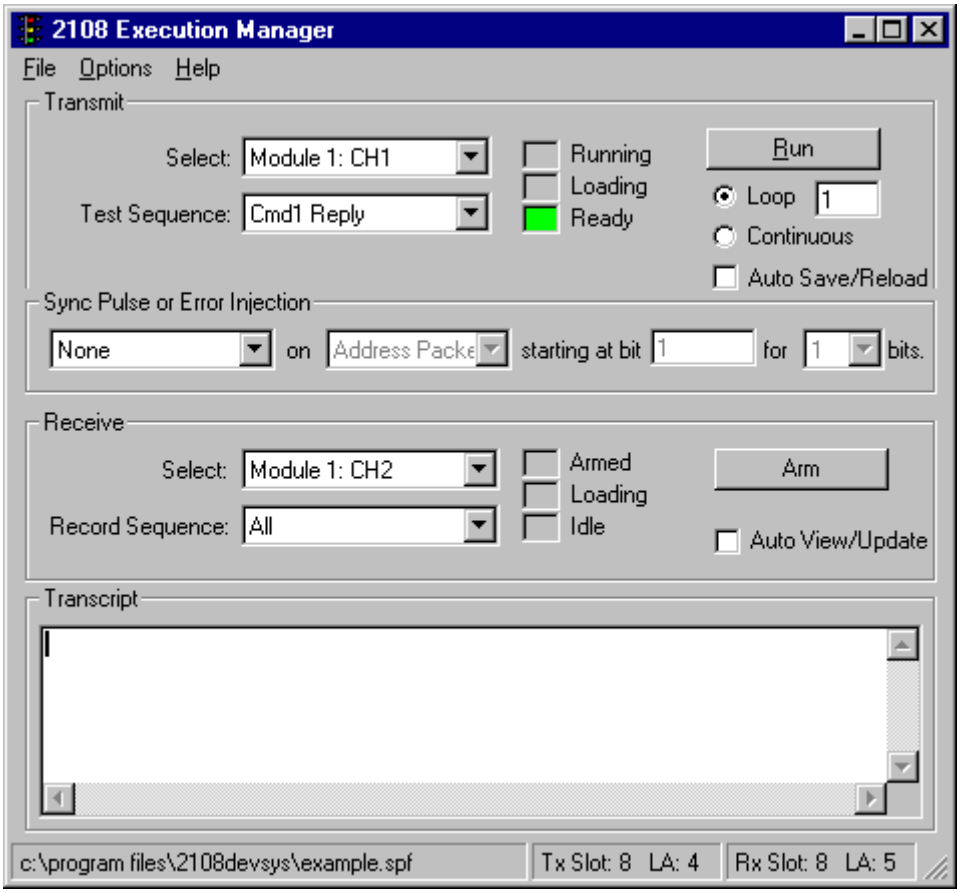
# 4.6 Receiver VXI Triggers & Interrupts

The Receiver VXI Triggers & Interrupts window controls the source and routing for Trigger A. Trigger A may be driven by VXI triggers (0-3) or Trigger B from other channels within the 2108 module. This window also selects a signal for the VXI interrupt bus.



# 5. Execution Manager

The 2108 Execution Manager is one of the components of the 2108 Development System. The Execution Manager handles loading, executing and reporting test results for project files created with the 2108 Development Environment. The Execution Manager is used with the 2108 Development System during debug and it also runs standalone for production testing.




To use the 2108 Execution Manager simply load the project file, select the appropriate Transmit and Receive controls and press the Arm or Run buttons. Operating results and any error conditions are reported in the Transcript window.

The 2108 Execution Manager is designed to simultaneously control one transmit and one receive channel. The 2108 Execution Manager window contains a Transmit section, Receive section and a Transcript window.

The Transmit section contains Select and Test Sequence combo boxes, a group of status LED's and run controls. After loading a project file, the Select combo box lists the available 2108 Module transmit channels programmed into the project file. The Test Sequence combo box lists the test sequence names that are associated with the selected channel.

Pressing the Run button in the Transmit section causes the 2108 to run the selected Test Sequence on the currently selected Module and Channel hardware. While running, the button is labeled Stop. Pressing the Stop button immediately halts the 2108. The Stop button is always used to halt the 2108 when it is running in Continuous mode. The Loop value defaults to one, but it can be set to a count of 999.

The Auto Save/Reload check box controls the automatic reload of the project file. If the Auto Save/Reload check box is checked, the project file is automatically reloaded when the Run button is pressed if the software detects that a reload is necessary. If the Auto Save/Reload check box is unchecked, the operator is prompted for a decision to reload the project file.

There are three status LED's labeled Running, Loading and Ready in the Transmit section. Initially, all three LED's are grayed out. Once a project file is loaded, the Ready light turns green. While loading a project file, the Loading LED turns yellow. The Running LED turns yellow while the selected 2108 Transmit channel is actively running.

The Sync Pulse or Error Injection section is associated with the currently selected transmit Module and Channel. Selecting None in the far left drop down combo box disables this feature and the remaining controls for Sync or Error Injection are grayed out. A Sync pulse or Error Injection selection enables this control and makes packet, start bit and duration user selectable. If selected, a Sync Pulse or an Error will be injected within the selected Transmit sequence when the Run button is pressed.

The Receive section contains Select and Test Sequence combo boxes, a group of status LED's and run controls. After loading a project file, the Select combo box lists the available 2108 Module receiver channels programmed into the project file. The Record Sequence combo box lists the test sequence names that are associated with the selected channel.

Pressing the Arm button conditions the selected receive channel to receive data. The receive data is monitored for a Record Sequence pattern match. While the receiver is monitoring data, the Arm button is labeled Stop. When a receive match condition is encountered, the receiver goes to the Idle mode and the Stop button reverts to Arm. If the Auto View/Update box is checked, the receive data is automatically displayed by the Logic Analyzer.

There are three status LED's labeled Armed, Loading and Idle in the Receive section. Initially, all three LED's are grayed out. While loading a project file, the Loading LED turns yellow. The Armed LED turns yellow while the selected receiver module is actively monitoring receive data. Once a Record Sequence condition is met, the Idle LED turns green indicating that the receiver has entered an idle state.

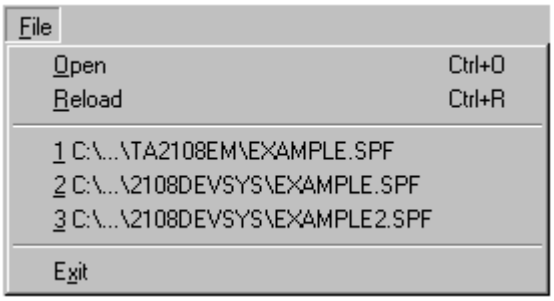
The Transcript window is a continuous journal of the test activity and test results. Its contents can be copied, pasted or cleared with the Options Menu. When the Transcript window buffer is full, a portion of the oldest data is deleted. During normal operation, the Transcript window logs start, end and operating status. The Transcript window also displays any error messages reported by the system.

The status bar at the bottom of the 2108 Execution Manager window shows the currently loaded project file and Slot and Logical Address for the selected transmit and receive modules.

# 5.1 Menus

## 5.1.1 File Menu

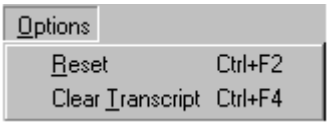
The Execution Manager File Menu supports the selection and management of 2108 project files.




| Menu Option | Description |
|---|---|
| **Open** | Opens a file browser to select a project file to download to the 2108 hardware. |
| **Reload** | Reloads current project file and updates the 2108 hardware. |
| **File History** | This provides a quick selection from the last four project files loaded. The file names are stored in a state file (TA2108EM.INI). |
| **Exit** | Closes the 2108 Execution Manager. |

## 5.1.2 Options Menu

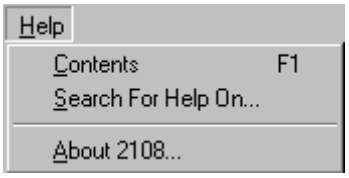
The Execution Manager Options Menu provides additional control over the user interface and the 2108 hardware.




| Menu Option | Description |
|---|---|
| **Reset** | Causes a hard reset of the 2108. This initializes it to a power-on state. |
| **Clear Transcript** | Clears the contents of the Transcript Window. |

## 5.1.3 Help Menu

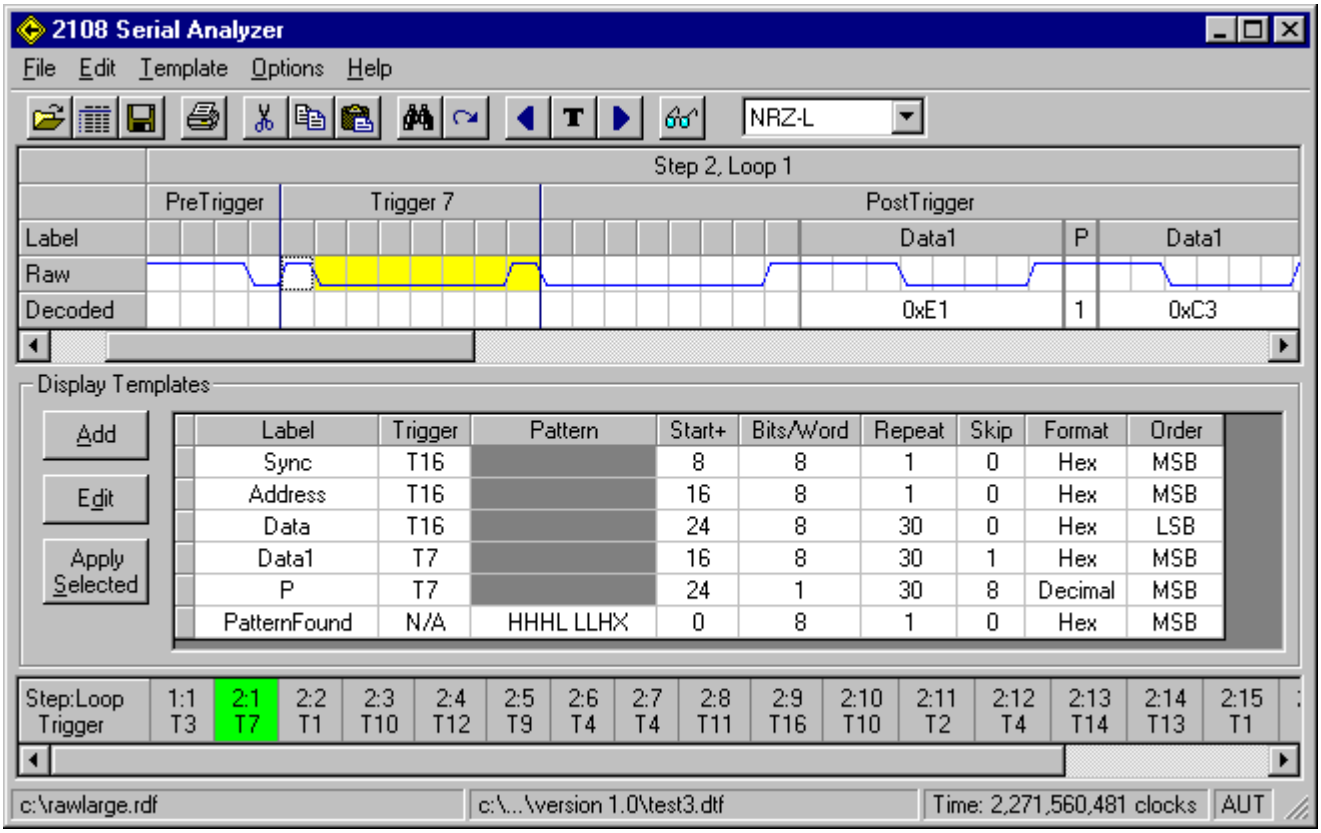
The Execution Manager Help Menu provides access to the help file and application version information. The help file is most easily accessed by pressing the **F1** function key. Using **F1** activates the help file topic that relates to the current window or dialog box.




| Menu Option | Description |
|---|---|
| **Contents** | Displays the Contents page of the help file. Function key **F1** also performs this task. |
| **Search for Help On…** | Displays the Search index of the help file. Use this to locate specific information in the help file. |
| **About 2108 Execution Manager…** | Displays a dialog containing version information and a copyright notice. |

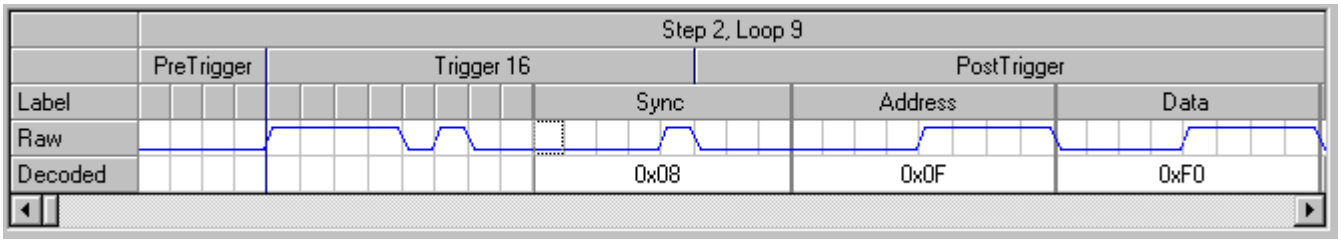
# 6. 2108 Serial Analyzer

The 2108 Serial Analyzer provides an interactive display to aid the designer or programmer in viewing and analyzing the data recorded by the 2108RX hardware. Recorded data may be viewed as “raw” data (waveforms without regard to format characteristics), or templates may be defined to add labels and decode the data into meaningful formats. Data streams may be searched based on patterns or trigger steps.



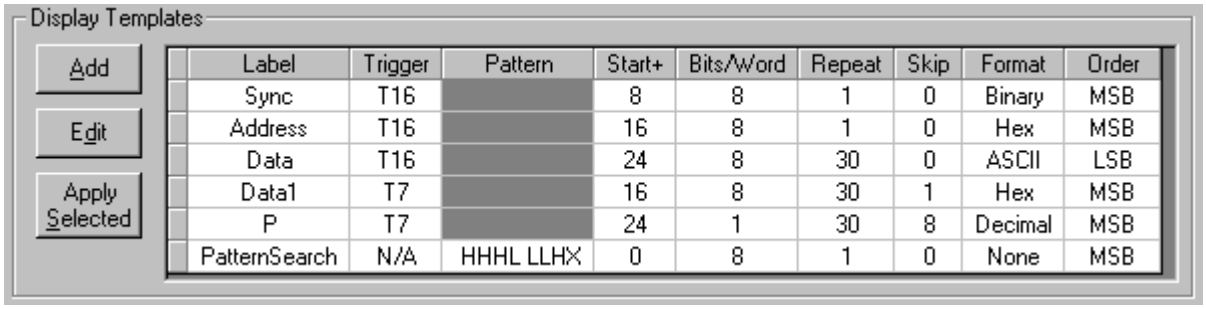
# 6.1 Receive Data

The recorded data (Raw), is displayed as a waveform based on one bit per clock regardless of the data format (i.e. a bi-phase data bit is displayed as two bits). Templates may be defined to group data bits and apply a meaningful label. The grouped data may also be displayed as Decoded data and expressed as decimal, binary, hex or ASCII.



# 6.2 Display Templates

The Display Templates pane displays the Label(s) and parameters associated with the Label(s). Parameters include the start sample position (trigger or pattern plus sample position), number of bits comprising the word, number of times the word is repeated and the number of samples to be skipped between words. Each word may be displayed as Binary, Decimal, Hex or ASCII. Bit order may be LSB or MSB.




| Label | Trigger | Pattern | Start+ | Bits/Word | Repeat | Skip | Format | Order |
|---|---|---|---|---|---|---|---|---|
| Sync | T16 | | 8 | 8 | 1 | 0 | Binary | MSB |
| Address | T16 | | 16 | 8 | 1 | 0 | Hex | MSB |
| Data | T16 | | 24 | 8 | 30 | 0 | ASCII | LSB |
| Data1 | T7 | | 16 | 8 | 30 | 1 | Hex | MSB |
| P | T7 | | 24 | 1 | 30 | 8 | Decimal | MSB |
| PatternSearch | N/A | HHHL LLHX | 0 | 8 | 1 | 0 | None | MSB |

Individual and multiple template entries can be cut, copied, pasted and deleted via the toolbar or Edit Menu. Copied entries can be pasted in other templates. Select a single entry with a left click in the leftmost column. Select multiple rows by drag-selecting in the leftmost column. The Label and Pattern columns can be individually selected and copied or pasted.

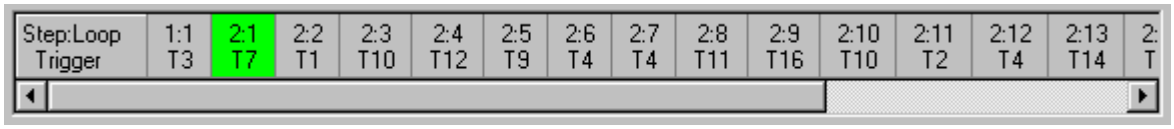
The Add button activates the Add Display Template dialog for creating a new template entry. Once defined, the new entry is placed at the bottom of the template display. The Edit button activates the same dialog to allow editing of the currently selected entry. Alternately, a double left click in the leftmost column does the same. Template entries may also be modified directly from the grid display.

The Apply Selected button allows selected template entries to be applied to the displayed data. This can be useful as a way to “debug” template behavior and ordering. Typically you would first reset all formatting via the Template Menu. A left click in the upper left corner of the grid selects all template entries.

> **Note: Template formatting is always reapplied when a new Step:Loop is selected from the Record Sequence list.**

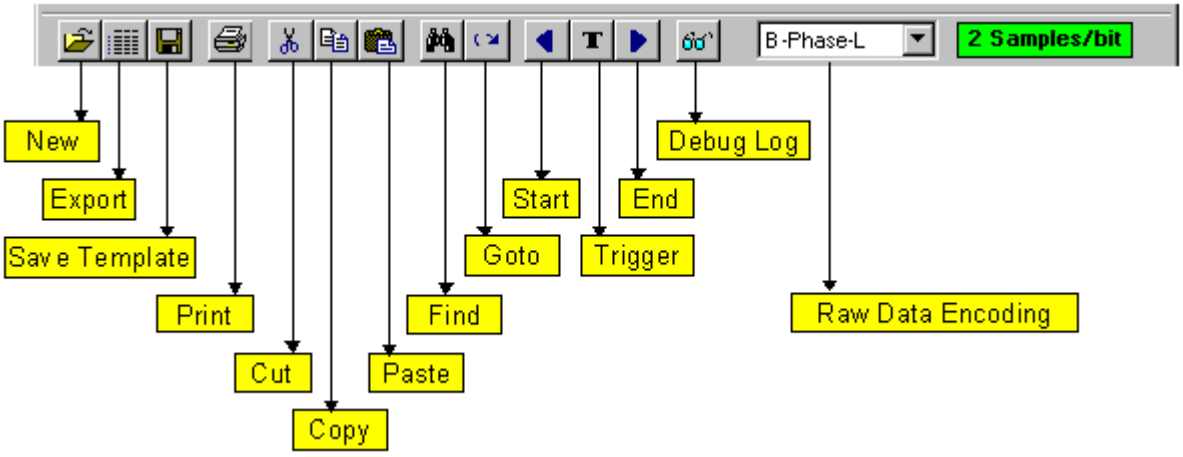
## 6.3 Record Sequence

The Record Sequence list provides a summary of the data recorded by the 2108RX hardware. The recording trigger conditions and loop counts are defined by the Record Sequences window of the 2108 Development Environment. A left mouse click on a Step:Loop entry causes it to be displayed in the Receive Data pane and overlaid with any relevant display templates. The selected step is highlighted with a green background. The clock count from the previous trigger is displayed when the mouse cursor is held over a Step:Loop entry for approximately two seconds.



# 6.4 Toolbar

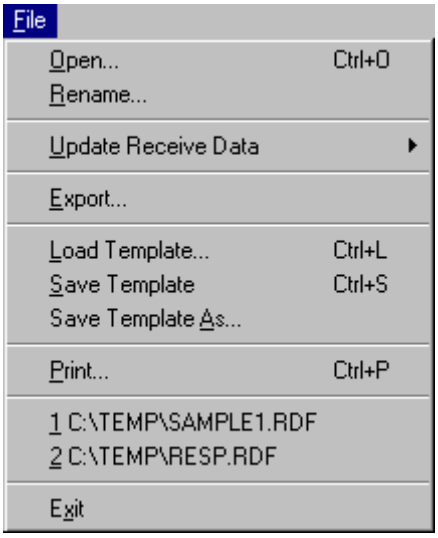
The toolbar on the 2108 Serial Analyzer provides quick access to commonly used commands. All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function. Note that the **Raw Data Encoding** selection determines how the receive data is decoded, displayed and searched.



# 6.5 Menus

## 6.5.1 File Menu

The 2108 Serial Analyzer File Menu is used to manage the opening of recorded data files (*.rdf) and the loading and saving of Template files (*.dtf). With this menu, 2108RX hardware recorded data and template files are opened, exported, created, loaded, and saved. A file history list permits quick reloading of recently accessed data files. The print command sends a screen capture of the application to the selected printer. The Serial Analyzer application can also be closed from this menu.

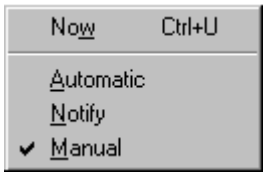


| Menu Option | Description |
|---|---|
| **Open…** | Opens a file browser for choosing a receive data file. The chosen file is loaded and displayed in the Raw data line. |
| **Rename…** | Renames current receive data file. |
| **Update Receive Data…** | Presents a submenu of options for updating the display of receive data. |
| **Export…** | Calls the Export Wizard which creates a delimited text file of specified record sequence steps. |
| **Load Template…** | Opens a file browser to select an existing template file to be applied to the data file being analyzed. |
| **Save Template** | Updates the template file with the latest editing changes. |
| **Save Template As…** | Creates a new template file with the latest editing changes. It then becomes the current template file. |
| **Print…** | Provides a screen print of the current display. |
| **File History** | This provides a quick selection from the last four RDF files loaded. The list is updated each time a new file is read. The file names are stored in a state file (2108SerialAnalyzer.ini). |
| **Exit** | Closes the 2108 Serial Analyzer. If the current template has not been saved, the user is prompted to do so before closing. |

When a template file is loaded, or created, a working copy is made to support interactive editing. When a template is saved, the original file is replaced with the working copy.

## 6.5.2 Update Receive Data Menu

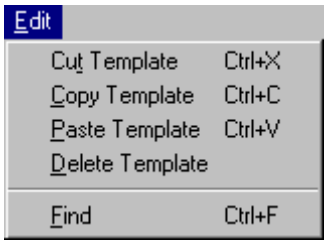
The Update Receive Data Menu, a submenu to the 2108 Serial Analyzer File Menu, is used to control the update behavior of the receive data display. This provides selections to automatically update the data when recorded by the receiver (in conjunction with the Execution Manager), notify the user when new data has been recorded or have updates be under the user's control.




| Menu Option | Description |
|---|---|
| **Now** | Reload the current receive data file. This is only active when the file has been modified since its last load. Also accomplished by pressing **Ctrl+U**. |
| **Automatic** | Reload and update the display whenever the receive data file is modified. |
| **Notify** | Notify the operator whenever the receive data file is modified. |
| **Manual** | Let the operator have complete control over reloading receive data for display. |

## 6.5.3 Edit Menu

The 2108 Serial Analyzer Edit Menu is used to manage template parameters. It controls cutting, copying, pasting, and deleting templates. Highlight the template row to be edited by clicking on the leftmost column or drag the cursor to select multiple template rows. The Edit menu also provides a Find feature to search the receive data for items based on Raw samples, Decoded data or Labels. Note that the **Find** option activates the Find dialog automatically loaded with the selected raw data, decoded data or label.



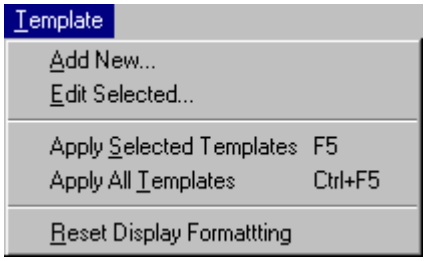

| Menu Option | Description |
|---|---|
| **Cut Template** | Removes the highlighted template(s) from the grid and stores them in the paste buffer for optional insertion elsewhere. |
| **Copy Template** | Copies the highlighted template(s) and places them in the paste buffer. |
| **Paste Template** | Inserts the contents of the paste buffer at the selected template row. |
| **Delete Template** | Deletes the highlighted template(s). |
| **Find…** | Activates a Find dialog to search for selected data patterns or labels. |

Individual items and columns can also be cut, copied and pasted with the Edit menu. Note that the Edit Menu captions change when columns or individual items are selected.

## 6.5.4 Template Menu

The 2108 Serial Analyzer Template Menu is used to create, edit or apply display templates. Several of these functions are also available as buttons on the Display Templates window. Note that the **Add New** option activates the Add Display Template dialog automatically loaded with the selected raw data and its position. If errors or warnings occur while applying a template, they are displayed in the Debug Log window.

**Note: Template formatting is always reapplied when a new Step:Loop is selected from the Record Sequence list.**




| Menu Option | Description |
|---|---|
| **Add New…** | Calls the Add Display Template dialog to add a new template entry to the bottom of the template list. |
| **Edit Selected…** | Calls the Edit Display Template dialog to edit the currently selected template row. |
| **Apply Selected Templates** | Applies the highlighted display template entries to the Raw data. Also achieved with **F5** and the Apply Selected button. |
| **Apply All Templates** | Resets formatting and reapplies all the display templates to the Raw data. Also achieved with **Ctrl+F5**. |
| **Reset Display Formatting** | Removes all formatting from the data display. |

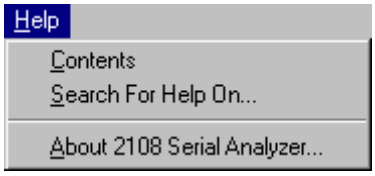
## 6.5.5 Options Menu

The 2108 Serial Analyzer Options Menu is used to step through the various sequence record steps and to move within the currently displayed sequence step. Note that many of the menu items may also be activated by shortcut keys.


| Menu Option | Description |
|---|---|
| **Next Step** | Displays the next record sequence step. Also achieved with **Ctrl+N**. |
| **Back Step** | Displays the previous record sequence step. Also achieved with **Ctrl+B**. |
| **Goto…** | Jumps to a specified position in the displayed record sequence step. The position is defined relative (+/-) to the trigger. Also achieved with **Ctrl+G**. |
| **Step Start** | Jumps to the start of data for the displayed record sequence step (i.e. sample one of pre-trigger). |
| **Step Trigger** | Jumps to the trigger position within the displayed record sequence step. Also achieved with **Ctrl+T**. |
| **Step End** | Jumps to last sample position of the displayed record sequence step. |
| **Debug Log** | Activates the Debug Log window which provides a list of errors incurred while applying templates. Also achieved with **F3**. |

## 6.5.6 Help Menu

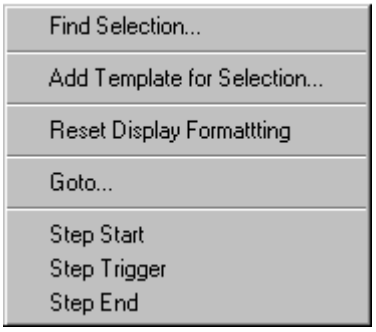
The 2108 Serial Analyzer Help Menu provides access to the help file and application version information. The help file is most easily accessed by pressing the **F1** function key. Using **F1** activates the help file topic that relates to the current window or dialog box.




| Menu Option | Description |
|---|---|
| **Contents** | Displays the Contents page of the help file. Function key **F1** also performs this task. |
| **Search for Help On…** | Displays the Search index of the help file. Use this to locate specific information in the help file. |
| **About 2108 Serial Analyzer…** | Displays a dialog containing version information and a copyright notice. |

## 6.5.7 Pop-up Menu

The 2108 Serial Analyzer Pop-up Menu is activated by right clicking in the Receive Data display area. The menu provides access to data search, display and formatting capabilities. Note that the **Add Template for Selection** option activates the Add Display Template dialog preloaded with the selected raw data and its position.




| Menu Option | Description |
|---|---|
| **Find Selection…** | Activates the Find dialog to search for the selected raw data, decoded data or label. The search parameters are preloaded with the selected data. |
| **Add Template for Selection…** | Calls the Add Display Template dialog to add a template based on the current data selection. |
| **Reset Display Formatting…** | Removes all formatting from the data display. |
| **Goto…** | Jumps to a specified position in the displayed record sequence step. The position is defined relative (+/-) to the trigger. Also achieved with **Ctrl+G**. |
| **Step Start…** | Jumps to the start of data for the displayed record sequence step (i.e. sample one of pre-trigger). |
| **Step Trigger…** | Jumps to the trigger position within the displayed record sequence step. Also achieved with **Ctrl+T**. |
| **Step End…** | Jumps to last sample position of the displayed record sequence step. |

# 6.6 Dialogs & Windows

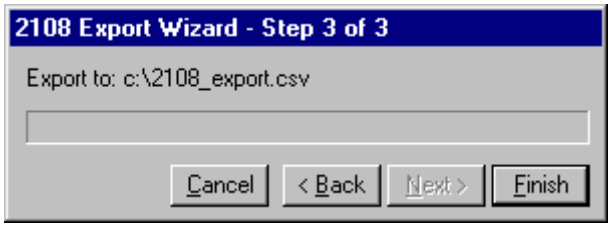
## 6.6.1 Export Wizard

The Serial Analyzer Export Wizard is a way to create a delimited text file of specified receive data. The resulting file can then be imported into a spreadsheet, database or other data analysis software tools. The export operation is accomplished in a 3 step process:

**Step 1: Select the data to be exported -**

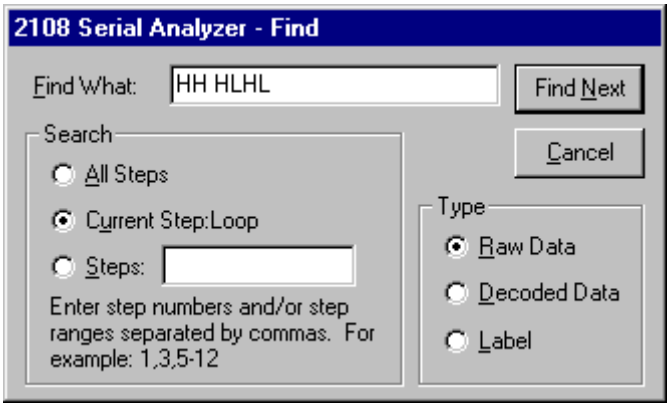

**Step 2: Specify file name, location and type (i.e. comma delimited or tab delimited) for storing the data -**

**Step 3: Confirm export and display operation status (Back returns to first step) -**



## 6.6.2 Serial Analyzer Find

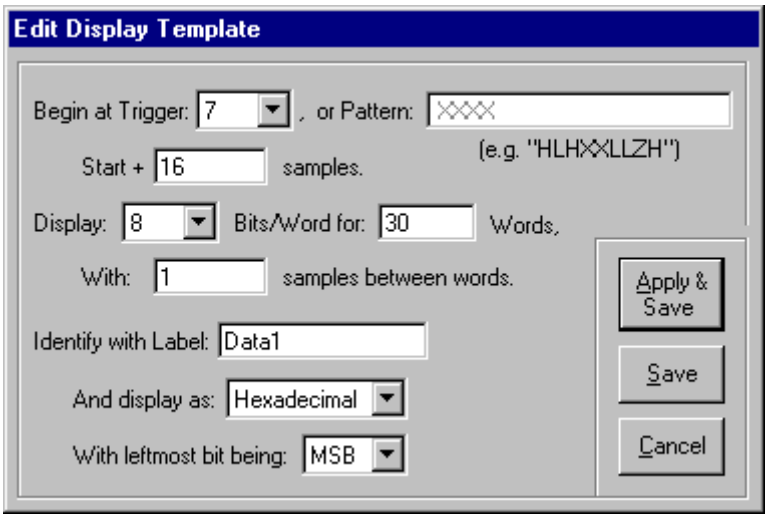
The Serial Analyzer Find dialog is used to search receive data. Searches are performed on raw data patterns, decoded data values or labels. Searches may be on all or selected record sequence steps. The Find dialog "floats" above other windows and remains active until it is canceled. When activated, from the Edit Menu or Pop-up Menu, the Find dialog automatically creates a search pattern from the currently selected raw data, decoded data or label (select raw and decoded data to choose label for search).



## 6.6.3 Add/Edit Display Template

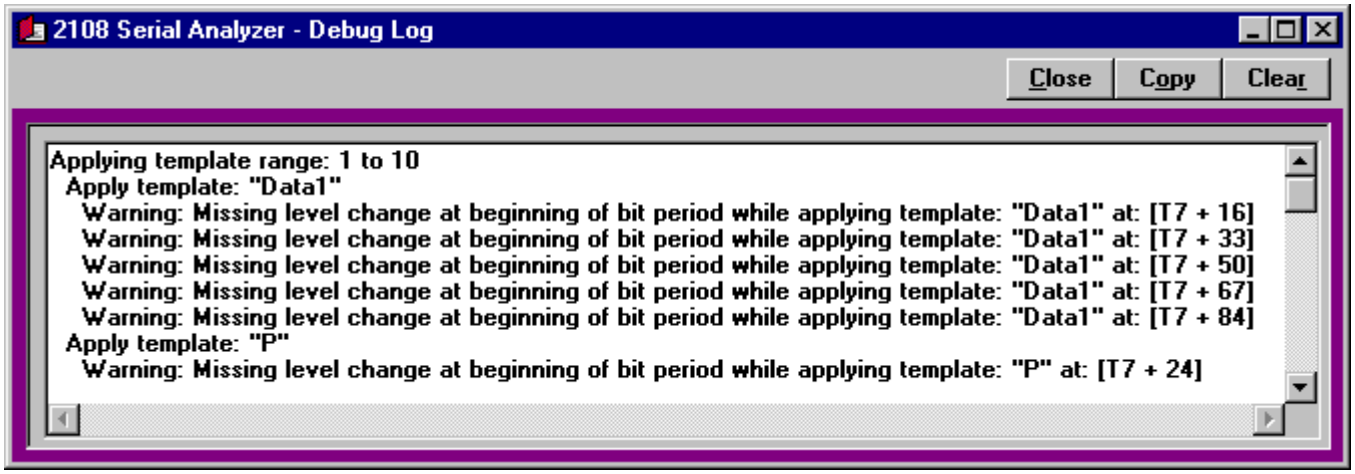
The Edit Display Template dialog is used to create or edit templates. It is activated from the Template Menu, Pop-up Menu or the Add button. Display templates are used to add formatting to the Receive Data. The display formatting is helpful for identifying data packets and their contents. Templates are applied based on the location of a trigger or specified data pattern. A relative position for the display field is specified as well as a width. Templates can be applied repeatedly with optionally skipped samples (e.g. to skip a parity bit). A Label and data display formatting are defined for the detected fields. Decoded data may be displayed as decimal, binary, hex or ASCII.

**Note: When adding a new template, the Add Display Template dialog automatically loads the currently selected raw data and position.**



## 6.6.4 Debug Log

The 2108 Serial Analyzer Debug Log window provides the user with a list of problems or potential errors encountered while applying templates to receive data.



# Index