OPERATORS MANUAL

Model 2108

VXI Serial Data System
Digital Resource Module

Manual Revision: 06/22/06
Manual Part Number: 2108RM003
Instrument Part Number:
30080/30150/30250

talon
INSTRUMENTS
An EADS North America Defense Company

CERTIFICATION

Talon Instruments certifies that this product met its published specifications at the time of shipment from the factory.

WARRANTY

Talon Instruments products are warranted against defects in materials and workmanship as follows:

(a) One year for the 2108 motherboard and all modules.

(b) Ninety days for cables and adapters.

During the warranty period, Talon Instruments will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to the Talon Instruments factory. Buyer shall prepay shipping charges to the factory and Talon Instruments shall pay shipping charges to return the product to the Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to Talon Instruments from another country.

Talon Instruments warrants that its software and firmware designated by Talon for use with its instruments will execute its programming instructions when properly installed on the instrument. Talon Instruments does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. TALON INSTRUMENTS SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. TALON INSTRUMENTS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

SAFETY FIRST

PROTECT YOURSELF AND THE EQUIPMENT.
Follow these precautions:

- Don’t bypass the chassis’ power cord’s ground lead with two-wire extension cords or plug adapters.
- Don't disconnect the green and yellow safety-earth-ground wire that connects the ground lug of the chassis power receptacle to the chassis ground terminal.
- Don't energize the chassis until directed to by the installation instructions.
- Don't repair the instrument unless you are a qualified electronics technician and have instructions from Talon Instruments.
- Pay attention to the **WARNING** statements. They point out situations that can cause injury or death.
- Pay attention to the **CAUTION** statements. They point out situations that can cause equipment damage.
- Use ESD static control procedures when handling the 2108 or any of its modules.

Table of Contents

List of Figures

List of Tables

1 Introduction

The Model 2108 is a flexible and capable serial bus emulator module that can be programmed to emulate a vast variety of serial bus protocols at speeds from 2Kbps to 200Mbps data rate. The flexibility of the 2108 module allows the user to fine-tune virtually all aspects of a serial bus interface, both at the electrical and logical levels.

The Model 2108 is comprised of a “C” size VXI motherboard which houses four instrument modules and four front-end signal conditioning modules. For serial emulation the instrument modules are the transmitter, 2108TX and the receiver, 2108RX . A number of interconnect modules for the 2108TX and 2108RX are available to provide signal conditioning for the various requirements imposed by different interfaces.

The Model 2108 may be configured with 1 to 4, 2108TX’s, 1 to 4, 2108RX’s or a combination of the two. If a 2108TX and 2108RX are installed in channel slots 1&2 or 3&4, they may be operated in a bi-directional mode.

1.1 Basic Elements of Serial Interfaces

There are a variety of serial interfaces being used today in the consumer, industrial and military sectors. Although a significant percentage of these interfaces are comprised of a number of standardized serial protocols, such as USB, IEEE 1394 and MIL-STD-1553, there is still an equally large percentage composed of custom serial interfaces. This group includes not only purpose-built interfaces but also variants of the standardized protocols.

In all instances, these standard and custom serial interfaces share some very basic elements. These elements include a set of electrical characteristics, a logical protocol and a method of clocking the data. In addition, some serial buses require special waveforms and control signals. The Model 2108 provides the resources in a programmable format to meet the physical and logical characteristics of most interfaces.

1.1.1 Serial Bus Electrical Characteristics

The data bit is the very building block of any serial transmission and assumes physical properties that must be defined to suit a particular protocol. The Model 2108 provides the following programmable resources to define the serial bus electrical characteristics:

A) **Signal Type**- differential, bi-polar or trinary

B) **Voltage Levels** - various levels from ECL, LVDS, TTL, +/-15Vdc, etc.

C) **Slew Rate** – min. 0.15V/ns to a max. 3V/ns

D) **Termination** – may be programmed on or off

The Model 2108 transmitter enables the user to set not only the default electrical characteristics of the bit, but also a second set which may be used to create “error” states.

1.1.2 Data Clocking

The Model 2108 provides internal clock sources for each channel which supports data rates from 2Kbps to 200Mbps. Additionally, the user may also select to clock the channels from external sources at the 2Kbps to 200Mbps rates. In addition the 2108RX receiver may be phase locked to the external clock or to the received data.

1.1.3 Waveform Data

Some serial buses require data which cannot be generated using logical ones and zeros to be output as part of a data stream. This data is usually referred to as “invalid bit format”. The Model 2108TX transmitter provides register space to define up to eight waveforms. Four can be inserted in serial data streams and four are used to generate word or frame gaps. The 2108RX receiver’s trigger logic may be programmed to recognize and trigger on waveform data as well.

1.1.4 Control Signals

In addition to clock and data signals some serial buses require separate signals for sync, enables and triggering. The Model 2108 provides additional signals for emulating complex serial interfaces or to synchronize with other instruments in the test system . These signals are:

Markers (2)	Output from the Model 2108TX, these signals may be programmed aligned or 1 to 3 bit times prior to or following the first and last data bits of a word or frame.
Output Flags (2)	Output from the Model 2108TX, these signals are pulses and can be used to trigger the UUT or other instruments in the test system. They are aligned to the first bit of a word or frame and remain true until the word is output.
Strobe (1)	Output from the Model 2108TX, this signal can be used to sync to the data bits or as a clock in bi-phase format applications.
Sync Pulse (1)	Output from the Model 2108TX, this signal may be programmed to start at a bit and remain true for up to 16 bits.
Input Flags (2)	Input to the Model 2108TX, these signals may be used to start execution of a data stream or word from the Model 2108TX Transmitter.
Trigger Data (5)	Signals routed from the 2108RX to the adjacent 2108TX as well as the front panel, used to start execution of a unique data table and monitor receiver status.
Qualifiers (2)	Input to the Model 2108RX, tests for High or Low to trigger recording.

2 Installation

The following sections describe the hardware and software installation procedure for the 2108. Refer to the “2108 Reference Manual” for module installation and jumper/termination options.

2.1 Hardware Installation

The following sections discuss the installation procedure for the 2108 system.

WARNING

Use ESD protocols whenever handling the 2108 or any of its modules.

2.1.1 Receiving Inspection

Check the shipment at the time of delivery and inspect each box for damage. Describe any box damage and list any shortages on the delivery invoice.

1. **Unpack the boxes.** Unpack the boxes in a clean and dry environment. Save all the packing material in case the instrument must be returned for service.

   In addition to the 2108 system, the following items are shipped with each unit:

   - 2108 Operators Manual
   - 2108 Reference Manual
   - 2108 Interconnect Reference Manual
   - 2108 Configuration Sheet
   - 2108 System CD (Includes 2108 VXIplug&play Instrument Driver, 2108 Development System and electronic copies of the manuals in PDF format.).

2. **Check for damage.** Inspect the equipment carefully for any signs of physical damage regardless of the condition of the shipping boxes. In the case of physical damage, call the shipper immediately and start the claim process. Call the Customer Service representative (800-722-2528) to inform them that the shipment arrived damaged. Please be prepared to provide a detailed damage report.

2.1.2 Returning Equipment

Follow these steps when you return equipment to Talon:

1. **Save the packing material.** Always return equipment in its original packing material and boxes. If you use inadequate material, you’ll be responsible for any shipping damage repair as carriers won’t accept responsibility on incorrectly packed equipment.
2. **Call Customer Service and ask for a return authorization.** The Customer Service representative (800-722-2528) will ask for your name, telephone number, company name, equipment type, model number, serial number, and a description of your problem.
3. **Pack and ship the equipment to:**

   Talon Instruments
   4 Goodyear
   Irvine, CA 92618

2.1.3 Preparation for Storage

The 2108 should be stored in a clean, dry environment. In high humidity environments, protect the 2108 from temperature variations that could cause internal condensation. The following environmental conditions apply to both shipping and storage:

Temperature	-40°C to +70°C
Relative Humidity	Not controlled, non-condensing
Altitude	<40000 ft. (12192 m)
Vibration	<2g
Shock	<40g

2.1.4 Preparation for Use

Prior to installing the 2108 module in the VXI chassis, several settings can be made via two switches located on the 2108 baseboard.

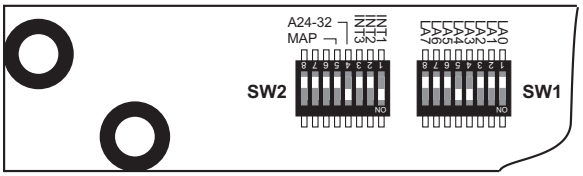


Figure 2-1 2108 Baseboard Rev C Switch Settings

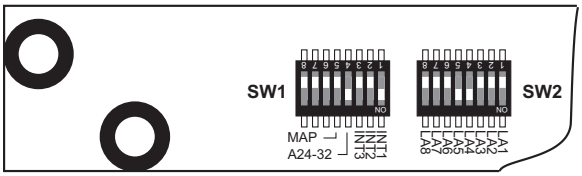


Figure 2-2 2108 Baseboard Rev B Switch Settings

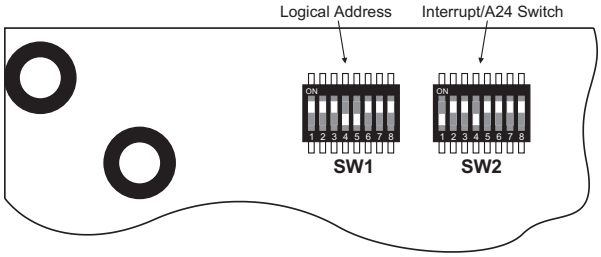


Figure 2-3 2108 Baseboard Rev NC Switch Settings

The baseboard revision is located on the back side etched in copper.

The following sections covers the following topics:

- Logical Address Selection
- VXI Interrupt Level Selection
- A24/A32 Map Selection

2.1.4.1 Logical Address Selection

The VXI chassis Resource Manager identifies units in the system by the unit's logical address.

The logical address of the 2108 can be statically or dynamically configured. An eight position DIP switch located on the baseboard, see figure 2-1, is used to assign the logical address. A logical address setting of 255 (all positions down, factory default) will enable the dynamic addressing mode. The logical address for each channel will be assigned by the resource manager. Logical address zero is reserved for the slot 0 controller and is invalid. Any other logical address setting will cause the 2108 to request four consecutive logical addresses starting with the address coded on logical address switch. The logical address setting must be a multiple of 4, the lower two logical address bits are ignored.

Logical Address Switch							
8	7	6	5	4	3	2	1
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0

Switch position 1 through 8 corresponds to bits 0 through 7 of the logical address. The "ON" setting sets the corresponding bit of the logical address to a one (1).

Channel one of each 2108 baseboard will be mapped to the lowest logical address assigned and channel four will be mapped to the highest.

The figures above shows an example where the logical address is set to 24 (position 5 and 4 down). This will cause the following logical address mapping:

Channel 1: LA24 (logical address 24)

Channel 2: LA25

Channel 3: LA26

Channel 4: LA27

The 2108 baseboard will be assigned four logical addresses even if less than four modules are installed.

2.1.4.2 VXI Interrupt Selection

The first three switch positions of Interrupt/Map switch are used to assign the VXI interrupt level. A value of zero disables VXI interrupt generation by the 2108. Values between one and seven select the interrupt of the same value, i.e., if position 2 and 3 are on and position one is off then VXI interrupt level 6 will be used by the 2108.

Factory default: VXI interrupts disabled.

Interrupt/Map Switch							
8	7	6	5	4	3	2	1
Reserved			Mapping		Interrupt Level		

Position 3	Position 2	Position 1	VXI Interrupt Level
OFF	OFF	OFF	Disabled (none)
OFF	OFF	ON	Level One Selected (factory default)
OFF	ON	OFF	Level Two Selected
OFF	ON	ON	Level Three Selected
ON	OFF	OFF	Level Four Selected
ON	OFF	ON	Level Five Selected
ON	ON	OFF	Level Six Selected
ON	ON	ON	Level Seven Selected

2.1.4.3 A24/A32 Map Selection

The memory space each 2108 channel will request can be set to either A24 (position 4 up) or to A32 (position 4 down).

The size of the A24/A32 memory requested can be set to either 4M (position 5 up) or 2M (position 5 down). Empty channel slots will not request any A24/A32 memory.

Factory Default: A32 memory selected, 4M map size.

Interrupt/Map Switch							
8	7	6	5	4	3	2	1
Reserved			4M/2M	A24/A32	Interrupts		

Position 5	Position 4	Register Mapping
OFF	OFF	A32 4M Register Mapping (factory default)
OFF	ON	A24 4M Register Mapping
ON	OFF	A32 2M Register Mapping (factory default)
ON	ON	A24 2M Register Mapping

ATTENTION

GPIB-VXI slot zero controllers do not support A32 register transfers. A24 register mapping must be selected for 2108 operation with these controllers.

2.1.5 Installation

The 2108 must be installed in a VXI mainframe in any slot except slot 0 (zero), which is reserved for the Resource Manager. Always check P1 and P2 for bent pins prior to installation. When inserting the 2108

into the mainframe, it should be gently rocked back and forth to seat the connectors into the backplane receptacles.

2.1.5.1 Initial Power-On

With the 2108 properly installed in a VXI chassis, turn on the external power supplies, if required, followed by the chassis power. Refer to the appropriate interconnect module reference manual for external power requirements.

2.2 Software Installation

The 2108 is shipped with a VXI Plug&Play Instrument Driver as well as the 2108 Project Development System. Both pieces of software are included on the CD shipped with the 2108 system. In addition to the software, the CD also contains all the operator and reference manuals for the 2108 in PDF format.

2.2.1 VXI Plug&Play Instrument Driver

The 2108 Instrument Driver links the communication interface and an application development environment. It provides a higher level, more abstract view of the instrument. It also provides ADE-specific information that supports the capabilities of the ADE, such as a graphical representation.

Some of the ADEs that this Instrument Drivers supports are listed below:

- Borland Turbo C/C++
- Agilent Technologies Agilent VEE
- Microsoft Visual Basic
- Microsoft Visual C/C++
- National Instruments LabVIEW
- National Instruments LabWindows/CVI

Included with the Instrument Driver is the Soft Front Panel. The soft front panel is a graphical user interface for the 2108. It is used to verify communications and functionality when the 2108 is first integrated into the system.

The 2108 VXI Plug&Play Instrument Driver requires that VISA be installed. Contact your Slot 0 Resource Manager manufacturer.

To install 2108 Plug&Play Instrument Driver:

1. Insert the 2108 System CD into your computer's CD-ROM drive. The 2108 System Installation menu will run automatically. If the installation menu, figure 2-4, does not appear, run setup.exe from the CD root directory.

Figure 2-4 2108 System CD Installation Menu

2. Press the “Install Instrument Driver” command button.
3. The installer will remove any versions that are currently loaded on the computer.
4. If a previous version was removed in step 3, then repeat step 2.
5. Follow the installer directions.
6. After the Instrument Driver is installed, the 2108 Soft Front Panel will be launched.

The following files are installed from the CD:

- ANSI C source code for the Instrument Driver and Soft Front Panel, i.e., .c, .h files.
- MS Windows 32 bit DLL library, i.e., ta2108_32.dll, ta2108.def files.
- Microsoft 32 bit DLL import library, i.e., ta2108.lib file.
- Function panel file, i.e., .fp file.
- MS Visual Basic Function Declaration text file, i.e., .bas file.
- Windows help file (.hlp file).

Visit the Talon Instruments web site at “www.taloninst.com” and check for 2108 Instrument Driver updates.

2.2.2 2108 Project Development System (PDS)

The 2108 PDS is a comprehensive application package that provides the user with easy to use graphical programming tools to develop set-up files, execute tests and view the recorded data. The package contains 3 distinct application programs to provide the majority of users with all the necessary functions to develop a functioning serial interface. It was developed for the Windows environment and operates on Win95, 98, 2000, or NT based systems.

2.2.2.1 Project Development Editor (PDE)

The Project Development Editor is the basic application program. It provides the resources to program the logical and physical characteristics of a serial interface. In addition it provides the tools to program data tables and test sequences for stimulating the Unit Under Test, (UUT). Record triggers based on unique start patterns are also easily programmed using the 2108RX receiver’s GUIs. Files created by the PDE are saved and may be downloaded using the VXI drivers or the Execution Manager.

2.2.2.2 Execution Manager (EM)

The Execution Manager application provides an interactive link to download set-up files from the VXI controller to a Model 2108. In addition it allows the user to select and run any test sequences defined using the PDE. The user may loop from 1 to 32k times or run in continuous mode to aid in program debugging. 2108RX receivers may be “ARMED” using the EM.

2.2.2.3 Serial Logic Analyzer (SLA)

The Serial Logic Analyzer application provides an interactive means of uploading recorded data from the 2108RX receiver. The data may be viewed as “raw” data or segmented by bit count, labeled and displayed as binary, hex, decimal or ASCII. This decoded data display is accomplished by defining templates that may be saved as a file. The template file may be applied to interactively recorded data when using the Execution Manager or used to analyze data recorded over a period of time. The SLA also provides search functions which operate on raw or decoded data or the label.

NOTE
Under Windows NT4, this software requires that Service Pack 2 or later be installed. Otherwise, the digital signals in the Timing and Table Editors may not display correctly.

To install 2108 Project Development System:

1. Remove any older versions of the Development System through the “Start->Settings->Control Panel->Add/Remove Programs”.
2. Insert the 2108 System CD into your computer’s CD-ROM drive. The 2108 System Installation menu will run automatically. If the installation menu does not appear run setup.exe from the CD root directory.
3. Press the “Install Development System” command button.
4. Follow the installer directions.

Visit the Talon Instruments web site at “www.taloninst.com” and check for 2108 Development System updates.

3 2108 Front Panel

The 2108 front panel provides the hardware interface to the UUT. Figure 3-1 illustrates the front panel and its connectors:

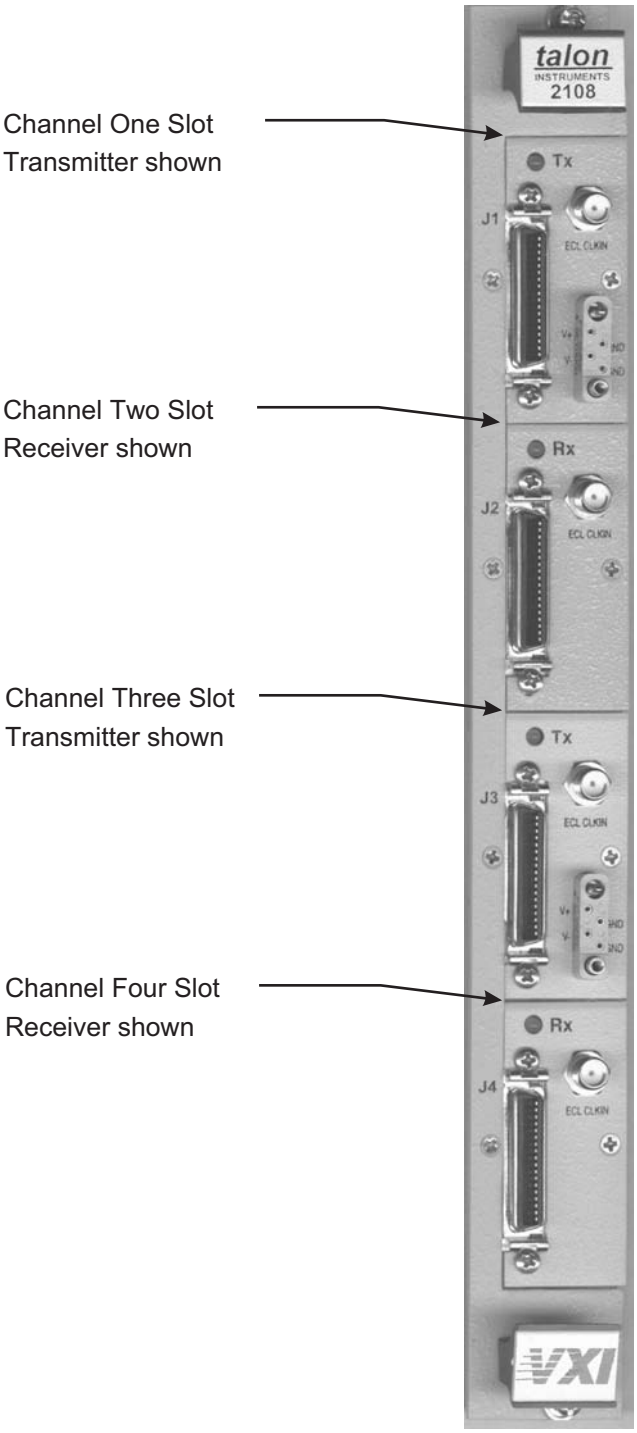


Figure 3-1 2108 Front Panel

The 2108 is a modular system that can hold up to four transmit or receive channels. Each transmit/receive channel connects to the UUT/ITA via the interconnect module. Refer to the specific interconnect module reference manual for detailed front panel description.

3.1 LED Indicators

The following sections describes the LED indicator for the transmitter and receiver interconnect modules.

3.1.1 Transmitter Interconnect LED Indicator

The transmitter interconnect module LED is used to indicate the operating status of the transmitter as well as the driver status. The color codes are:

OFF	The transmitter is outputting “Idle” or “Standby” CMT.
GREEN	The transmitter is outputting a user defined CMT.
RED	The drivers have shut down due to either an over current or over temperature condition.

3.1.2 Receiver Interconnect LED Indicator

The receiver interconnect module LED is used to indicate the operating status of the receiver. The color codes are:

AMBER	The receiver is currently waiting for a trigger (ARMed).
GREEN	The receiver is currently recording post trigger data.

3.2 SMA Connector

The SMA is used to input a single ended ECL clock (TxCLKIN1 or RxCLKIN1).

3.3 Power Connector

The transmitter power connector is used to provide the V+ and V- supply voltages to the programmable drivers.

Table 3-1 below lists the pinouts of the TX01 Power connector.

Pin Number	Signal
A	V+
B	GND
C	V-
D	GND

Table 3-1 Power Connector Pinouts

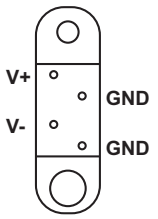


Figure 3-2 Power Connector

3.4 J1-J4 I/O Connector

The front panel I/O connector is used to connect the 2108 signals to the UUT/ITA fixture.

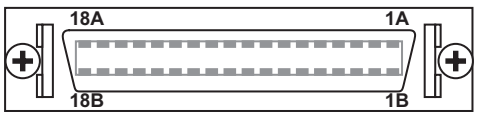


Figure 3-3 J1-J4 Connector

Table 3-3 lists the I/O channels assigned to J1-J4 for both the transmitter and receiver.

Pin Number	Transmitter		Receiver	
	Signal	Default Source	Signal	Default Source
2A	TxSig1	TxData	RxData+	
4A	TxSig2	TxClkOut	RxData-	
6A	TxSig3	TxMarker1	RxClkIn2+	
8A	TxSig4	TxMarker2	RxClkIn2-	
10A	TxSig5	TxFlagOut1	RxQual1+	
12A	TxSig6	TxFlagOut2	RxQual1-	
14A	TxSig7	TxSyncPulse	RxQual2+	
16A	TxSig8	TxBusy	RxQual2-	
18A	NU		RxSig1	RxArm
2B	TxFlagIn1		RxTrigValid	
4B	TxFlagIn2		RxTrigNum0	
6B	TxClkIn2+		RxTrigNum1	
8B	TxClkIn2-		RxTrigNum2	
10B	TxBusy		RxTrigNum3	
12B	TxSyncPulse		RxG0Val	
14B	NU		RxG1Val	
16B	NU		RxClkOut	
18B	NU		RxSig2	RxBusy
All other pins signal ground.				

Table 3-3 J1-J4 I/O Connector Assignments

3.5 2108 Front Panel Mating Connectors

The following table lists the manufacturer part number and Talon order number for the front panel mating connectors.

Table 3-2 lists the manufacturers part numbers and Talons part numbers for the 2108 mating connectors.

Connector	Manufacturer Part Number	Talon Order Number
J1-J4	AMP P/N "2-178238-5"	2108/300
Power	Positronix Industries P/N "SGM4FSCT0000"	2108/304

Table 3-2 Mating Connector Part Numbers

4 Operation

The steps required to create a test program using the 2108 are very similar to those required to use any other VXI module. Regardless of the user's choice of programming path - whether it is the plug&play instrument driver, Development Editor, A24/A32 register-based access or a combination of these - the following basic steps will apply:

1. INITIALIZE the VXI session for each 2108 channel.
2. CONFIGURE the 2108 settings and structures for the application.
3. Enable, or SOURCE, the output drivers.
4. Issue the EXECUTE commands.
5. Utilize STATUS and POST PROCESS functions to evaluate/analyze results.
6. CLOSE the VXI session.

The following sections describes each step in more detail with respect to the Development Editor and VXI plug&play instrument driver function(s).

Programming the 2108 using the Development System is described in the "2108 Development System Users Manual".

4.1 Initializing the VXI Session

Each channel of a 2108 module has a unique logical address and requires its own session handle.

If a project file is used to configure the 2108 channels then the "***ta2108_LoadProject***" function will initialize all the channels automatically by calling the "***ta2108_autoConnectToAll***" function. The session handle of any channel initialized by the "***ta2108_LoadProject***" function can be queried by first selecting the channel using the "***ta2108_Select***" function and then calling the "***ta2108_GetConfig***" function.

If VXI plug&play functions are used to configure the 2108 channels then any of the following functions can be used to open a session:

1. "***ta2108_init***"
2. "***ta2108_autoConnectToFirst***"
3. "***ta2108_autoConnectToLA***"
4. "***ta2108_autoConnectToSlot***"
5. "***ta2108_autoConnectToAll***"

4.2 Configure the 2108 Settings and Structures

Talon recommends using the Project Development Editor (PDE) to configure the 2108 channels. The PDE provides a graphical representation of the 2108 hardware resources.

The following sections describes the process for configuring the 2108 channels using the PDE. The corresponding VXI plug&play functions required to perform the same action will be listed as well.

4.2.1 Defining the 2108 PDE Configuration

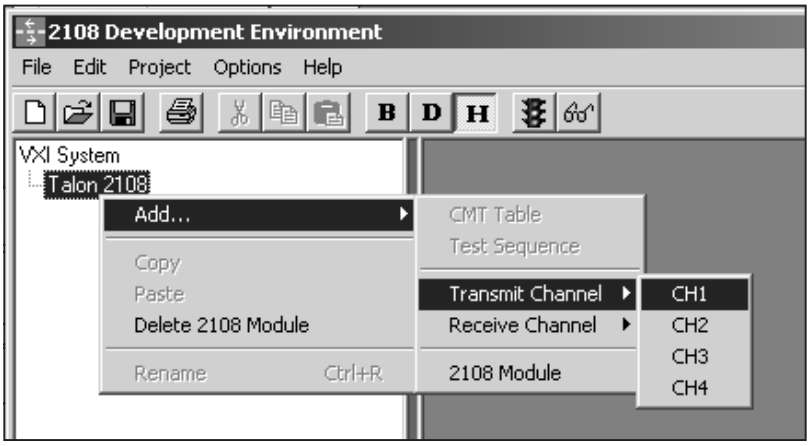
The first step in using the PDE is to configure the 2108 (s) being programmed by defining the number and type of instruments installed in each channel slot.

When started the PDE will load the last file that was last saved. To create a new project select the "**File->New**" command. The PDE will create a new project with one 2108 module that contains one transmitter channel and one receiver channel.

To delete a 2108 Module from the project, right click on the "**Talon 2108**" and select "**Delete 2108 Module**" command.

To add a 2108 Module, right click on "**VXI System**" in the menu directory and select "**Add...**", "**2108 Module**".

To add a transmit or receive channel, right click on the Talon 2108 module and select "**Add...**". Select the hardware and channel position. A configuration sheet is shipped with all Talon 2108 Modules that identifies the installed hardware.



This step defines the installed hardware for the PDE and does not have any corresponding VXI plug&play functions.

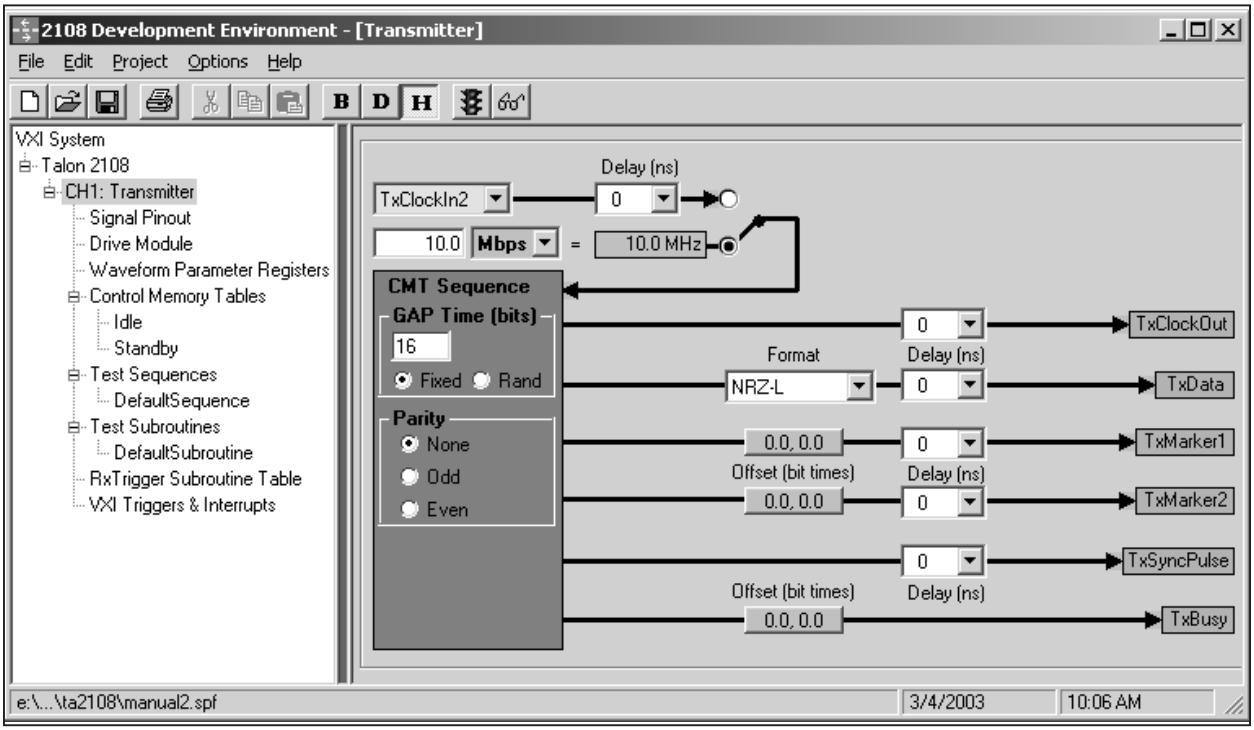
4.2.2 Configure the 2108 Transmitter

The 2108TX module provides the necessary resources to output data in a serial stream using different formats, protocols, data rates, etc. It also provides auxiliary signals such as a shared clock, markers, sync, triggers, etc., required by many serial interfaces.

The operation of the 2108TX requires the user to define the characteristics of the serial interface to be emulated. This includes data to be output and the sequence in which the user wants the data output.

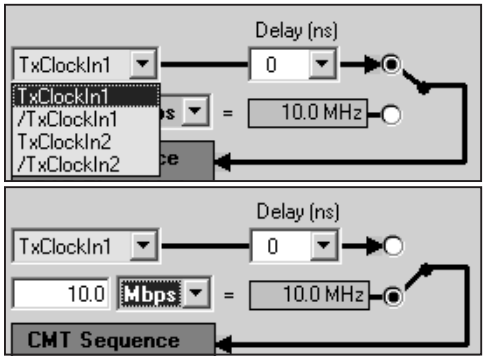
4.2.2.1 Logical Bit Settings

The logical bit characteristics are programmed using the Transmitter Panel shown below.

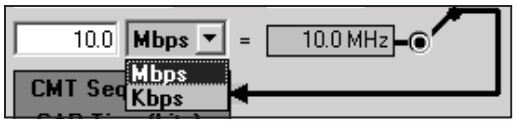


In the Transmitter Panel the user programs:

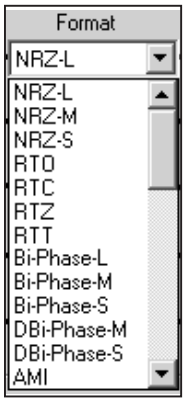
Clock Source External or internal clock sources may be selected as the transmitter master clock.



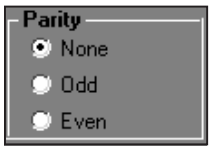
Bit Rate Bit rates from 2kBps to 200MBps may be entered.



Bit Formats The required bit format may be selected from a pull down selector.

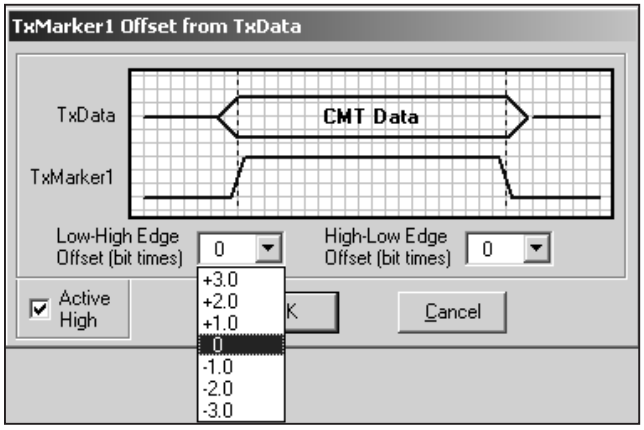


Parity Parity if required is set as None, Odd or Even.



Gap Gap times may be set to fixed or random with limits from 4 to 65k bit times.

Marker(s) May be set to operate coincident with the data or +/- 3 clocks from the active data and active high or low.



Delay Delays in 1ns increments from +/- 10ns may be programmed to most signals when cable or UUT delays cause synchronization problems.

VXI plug&play functions for logical bit programming:

ta2108_txSetSignalParameters

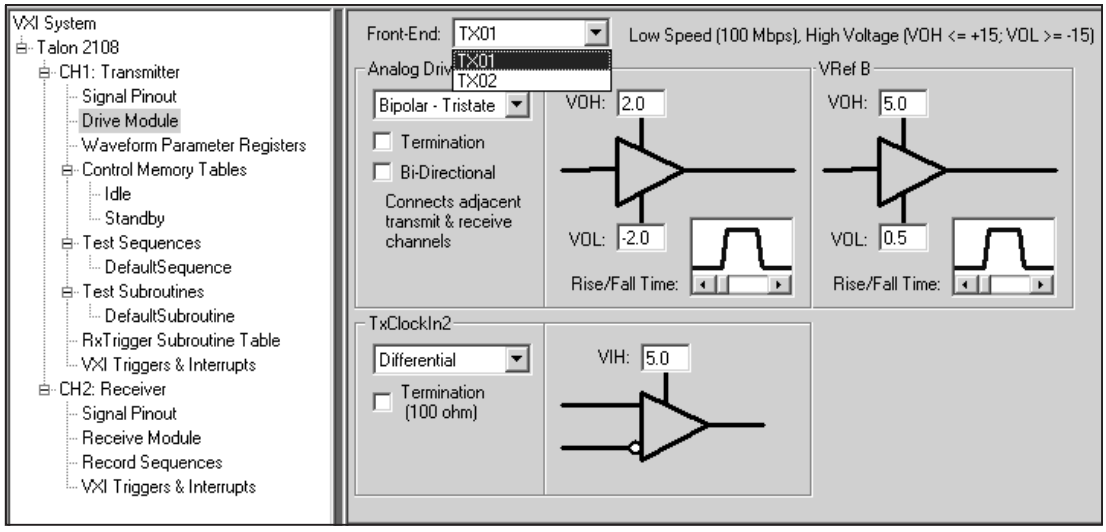
ta2108_txSetSignalOffset

ta2108_txSetSignalDelay

ta2108_txSetGapMinMax

4.2.2.2 Electrical Characteristics

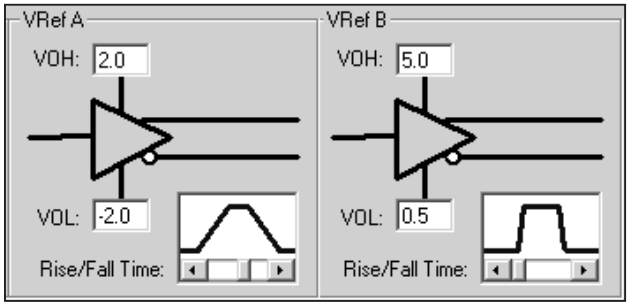
The electrical bit characteristics are programmed in the Drive Module Panel shown below. The front end interconnect type is also specified on this panel.

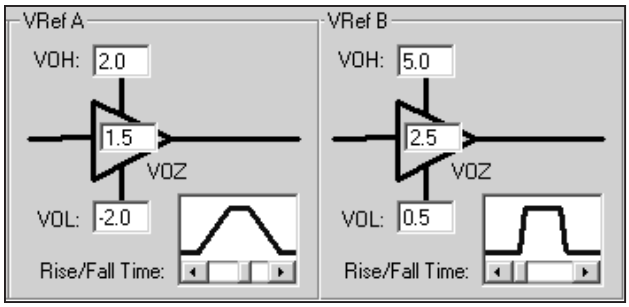


The bit characteristics to be programmed are:

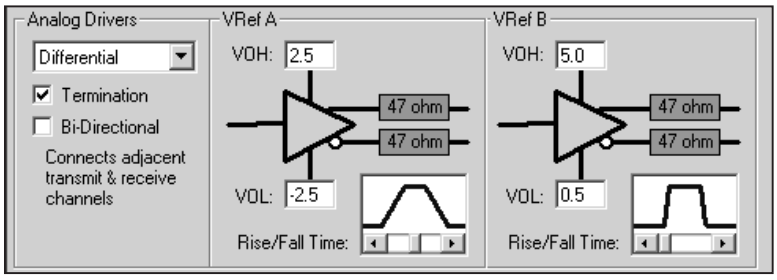
Signal Type The selections for most I/O modules include Differential, bi-polar with tri-state or bi-polar with a third state (trinary).

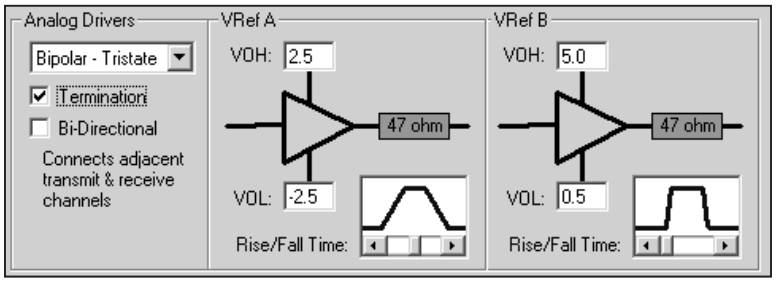
Signal Voltage VrefA is the primary voltage reference requiring the output levels VOH and VOL to be entered, (3rd voltage if Bi-polar VOZ is selected), slew rate is selected by use of the slide bar window (Rise/Fall Time); VRefB may be used as a secondary voltage reference to generate errors or as a margining voltage reference



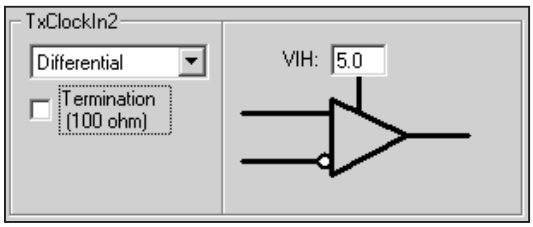


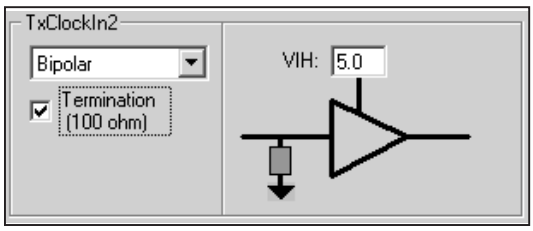
Termination Most of the I/O modules allow the selection of terminated or un-terminated signals.





TxClockIn2 If using external clock 2 as Clock Source the “signal good” level is programmed in this window for differential or bi-polar along with the termination choice.





Bi-directional If selected the transmitter I/O module will be the only connection to the interface bus and incoming data will be routed to the adjacent 2108RX module for bi-directional operations.

VXI plug&play functions for electrical bit characteristics:

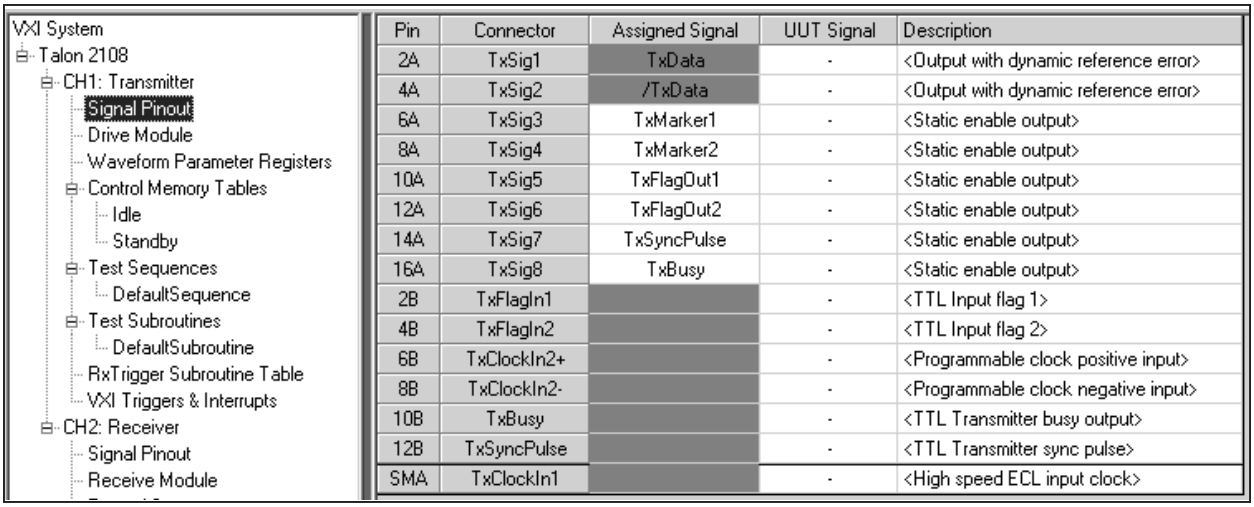
ta2108_txSetFrontEndControl

ta2108_txSetFrontEndInput

ta2108_txSetDriverReference

4.2.2.3 Signal Routing

The 2108TX has multiple signals which may or may not be needed for the serial interface being programmed.



Pin	Connector	Assigned Signal	UUT Signal	Description
2A	TxSig1	TxData	-	<Output with dynamic reference error>
4A	TxSig2	/TxData	-	<Output with dynamic reference error>
6A	TxSig3	TxMarker1	-	<Static enable output>
8A	TxSig4	TxMarker2	-	<Static enable output>
10A	TxSig5	TxFlagOut1	-	<Static enable output>
12A	TxSig6	TxFlagOut2	-	<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		-	<TTL Transmitter sync pulse>
SMA	TxClockIn1		-	<High speed ECL input clock>

The TxData and TxClockOut signals are always output on pins 2A and 4A if bi-polar mode is selected. If differential is selected then TxData will be on pin 2A and /TxData will be on pin 4A. Grounds will always be on the odd pins between signals.

The user selects the signals needed and routes them to the selected pins using the pulldown selections in the Assigned Signals column. The user may also elect to enter names in the UUT and Description columns for documentation purposes.

VXI plug&play functions for signal routing:

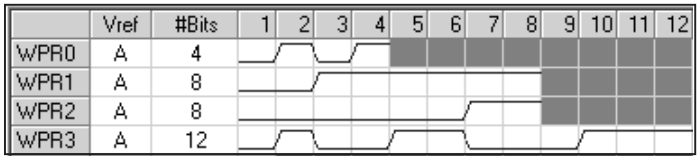
ta2108_txSetUserSignal

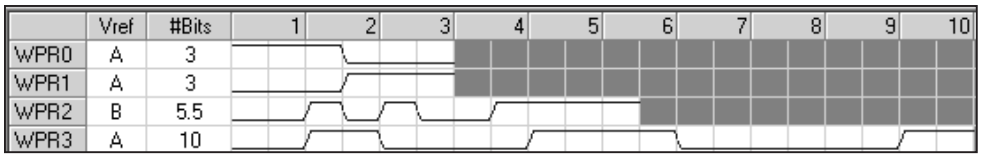
4.2.2.4 Waveform Definition

Some buses use non-logical bit patterns for sync codes or data. For example, the military 1553 bus has a command word sync code which is a signal represented as 1½ bit times high followed by 1½ bit times low. Another example is a pulse stream where 2 bits low out of 8 is a zero and 6 bits low out of 8 is a one. These waveforms can be programmed via the Waveform Parameters Register panel.

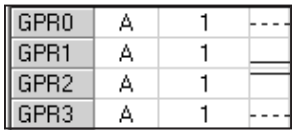
There are two types of waveform registers:

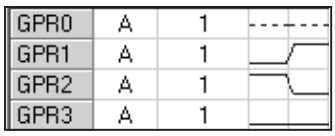
Waveforms Four custom waveforms (WPR0, WPR1, WPR3 & WPR4) may be defined as using Voltage Reference A or B, the number of bit times (4 to 12 bits), and the waveform. Waveforms are programmed by selecting a cell and clicking or entering an L or H to set desired state, (bi-phase selections allow ½ bit time selections between 2 & 12 bits).





Gap Waveforms There are four waveforms that may be used as Gap waveforms, (GRP0, GPR1, GPR2 & GPR3). They are represented as a single bit time and are programmed by selecting the Voltage Ref. A or B and the desired state, High, Low or Tri-state. Bi-phase selections allow for ½ bit time selections.





VXI plug&play functions for waveform definition:

ta2108_txSetWaveformRegisters

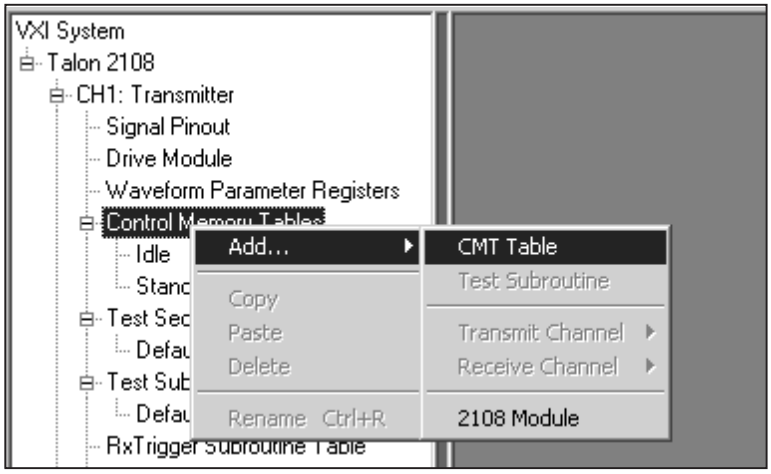
4.2.2.5 Define the Bit Sequence(s)

All serial interfaces pack the bits in some logical sequence such that the receiving device can reverse the sequence and extract meaningful data from the data stream. The schemes to logically pack data into “frames” or “packets” is unique for every interface. The 2108TX allows the programming of almost any conceivable combination of bit sequences, thereby allowing the emulation of interfaces with unique start/stop codes, parity bits, bit stuffing, as well as the generation of continuous data.

4.2.2.5.1 Control Memory Tables

The programming of bit sequences is accomplished in the Control Memory Table panel. This panel represents the 8Mbits of memory reserved for data to be transmitted by the 2108TX. Frames are defined as tables. A table may be just a data word or comprised of multiple data word(s), waveform(s), gaps or

random data words. New tables may be added by right clicking on Control Memory Tables in the menu directory, selecting Add CMT Table and entering a name for the table.



VXI plug&play functions for defining a CMT:

ta2108_txDefineCMT

4.2.2.5.2 Programming Control Memory Tables

A Control Memory Table is programmed by double clicking on the table name and entering or selecting the elements which define the table characteristics. The following entries are available:

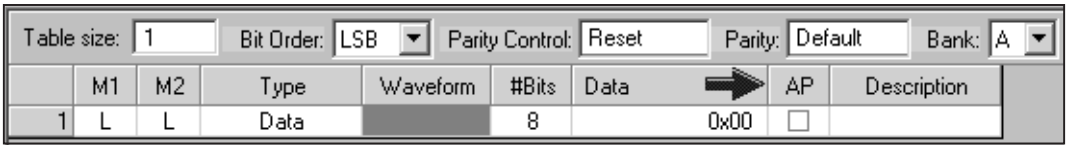
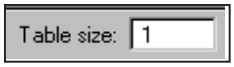
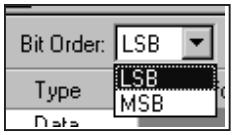


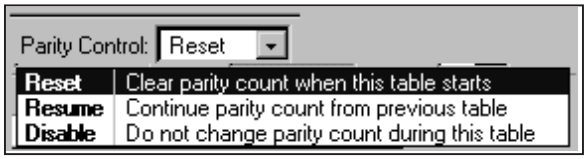
Table Size Enter the number of steps or words needed for the data. Any number up to the available memory depth may be entered.



Bit Order Select from the pull down window to select the order in which the data is transmitted, Most Significant Bit or Least Significant Bit.

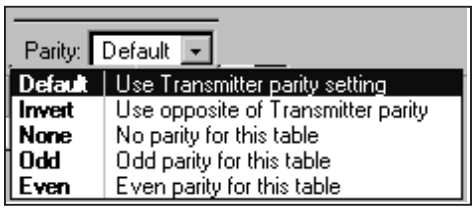


Parity Control Provides selections for the user to Reset the parity calculator, Resume to continue the parity calculation from a previous table, or Disable the parity calculation altogether.



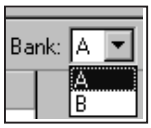
Parity

This window allows the user to select or override the default parity programmed in the Transmitter Panel.



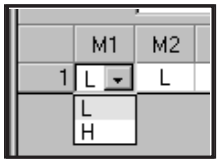
Bank

The 8Mbits of memory is divided into two 4Mbit banks for ping-pong operations, the pull-down selector is used to select the bank A or B to store the table being defined.



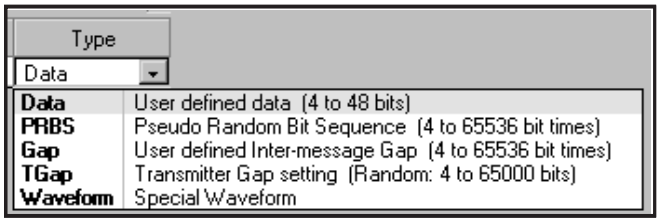
M1 or M2

If the Marker signals are to be used while the step data is transmitted the level is selected as low (L) or high (H) for M1 and/or M2.



Type

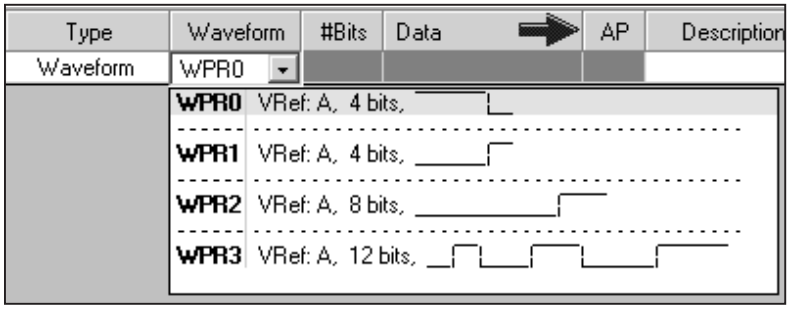
The type of data to be output for each step of the table is selected in this pull-down window by clicking on one of the following choices:

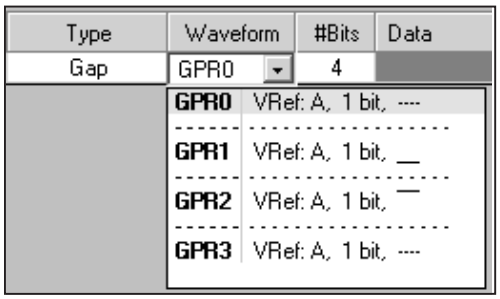


Data	Fixed data from 4 bits to 48 bits in length.
PRBS	Pseudo random bit sequence from 4 to 65k bits.
Gap	Fixed intermessage gap using one of the Gap Waveforms from 4 to 65k bit times.
TGap	Uses the default Transmitter Panel setting (random or fixed) and one of the Gap waveforms.
Waveform	One of the 4 waveforms defined in the Waveform Parameters Register panel will be selected.

Waveform

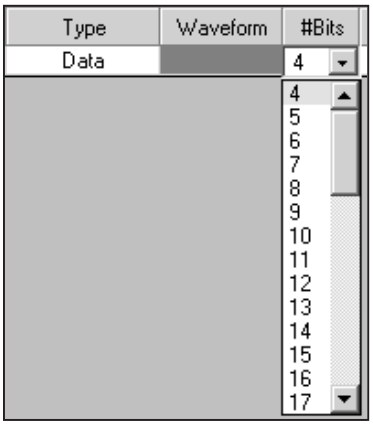
If Waveform, Gap or TGap was selected as the Type of data to be transmitted in a step this window will be active and allow the selection of one of the previously defined waveforms.



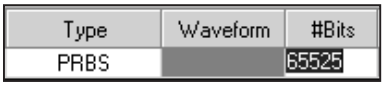


#Bits Allows the entry of the number of bits represented in a step:

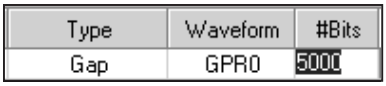
Data The number of bits is from 4 to 48 (words longer than 48 bits are entered in consecutive steps and will be output as a single stream).



PRBS The number of bits is from 4 to 65,536 of random data.



Gap The number of Gap periods is from 4 to 65,536 bit times.



Data Fixed data is entered in this window as binary, decimal or hex. Binary data is specified by specifying a 'b' at the end of the data, e.g., "1010 0001b". Decimal data does not require special characters. Hexadecimal data is specified by a "0x" preceding the data, e.g., "0x1234".
Spaces are allowed anywhere when entering data.
Once data in entered it will be displayed based on the selected format.

AP Clicking on this selection box will append a parity bit to the data using the parity control selected for this table, (parity may be calculated across any number of words by selecting AP only on the last data step in the table).

VXI plug&play functions for editing a CMT:

ta2108_txSelectCMTStep

ta2108_txInsertCMTStep

ta2108_txDeleteCMTStep

4.2.2.5.3 Examples of CMT's

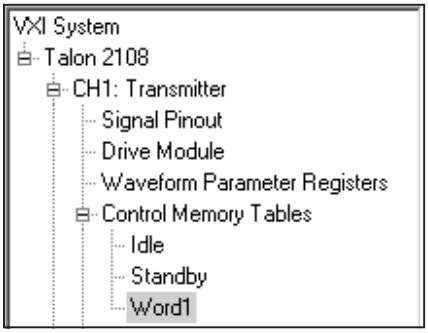
The following sections illustrates two CMT examples.

4.2.2.5.3.1 Simple 16-Bit One-Word Table

In this example, the user needs to output a single 16 bit, one word table, using M1 high as a write enable signal, MSB as the bit transmission order and adding a parity bit to the data. This type of data table is easily programmed by following these steps:

a) Right click on Control Memory Tables in the menu directory

b) Enter a name for the table, (Word1 for this example)



c) Leave 1 as the step size

d) Select MSB for the Bit Order

e) Select H for M1

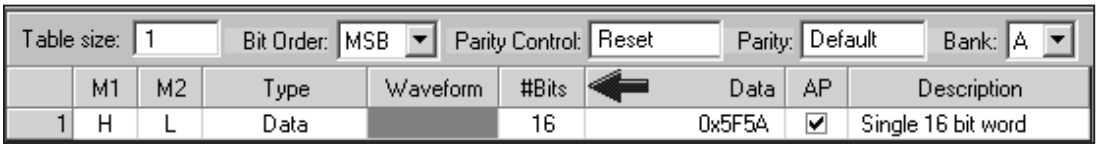
f) Select Data as the Type

g) Enter 16 as the #Bits

h) Enter the data (0101 1111 0101 1010b in this example)

i) Click on the AP box to add a parity bit

j) Add a description in the Description box if desired

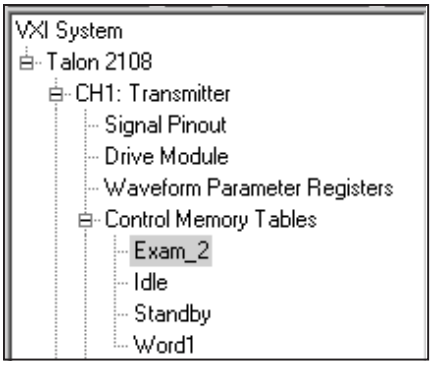


4.2.2.5.3.2 Two 40bit Words with Gap

In this example, the user wishes to output two 40bit words, (8 bit header followed by 32-bit data) separated by a 300 bit time fixed gap. Data is to be transmitted as LSB and parity is to be appended to the data. The steps are:

a) Right click on Control Memory Tables in the menu directory

b) Enter a name for the table, (Exam_2 for this example)



c) Enter 5 as the step size
d) Select LSB for the Bit Order
e) For Step 1 select Data as the Type
f) Enter 8 as the #Bits
g) Enter the data for header 1 (”0000 1111b” in this example)
h) For Step 2 select Data as the Type
i) Enter 32 as the #Bits
j) Enter the data for word 1 (”0x73CA 73CA” in this example)
k) Click on the AP box to append the parity bit
l) For Step 3 select Gap as the Type
m) Select GPR0 as the waveform to be transmitted as the Gap signal
n) Enter 300 as the #Bits
o) For Step 4 select Data as the Type
p) Enter 8 as the #Bits
q) Enter the data for header 1 (”0011 1100b” in this example)
r) For Step 5 select Data as the Type
s) Enter 32 as the #Bits
t) Enter the data for word 1 (”0x9160 9103” in this example)
u) Click on the AP box to append the parity bit

Table size: 5 Bit Order: LSB Parity Control: Reset Parity: Default Bank: A

	M1	M2	Type	Waveform	#Bits	Data	AP	Description
1	L	L	Data		8	0x0F	☐	Header
2	L	L	Data		32	0x73CA73CA	☑	Word 1
3	L	L	Gap	GPR0	300			Gap
4	L	L	Data		8	0x3C	☐	Header 2
5	L	L	Data		32	0x91609103	☑	Word 2

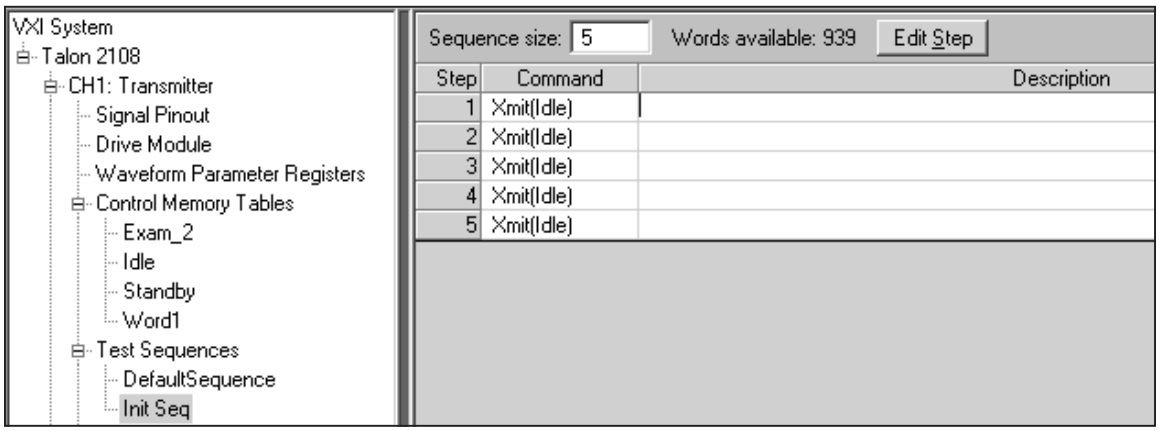
4.2.2.6 Define the Frame Output Sequence(s)

Once the bit format and data frames are defined, the 2108 allows the user to output these frames in any sequence he chooses. This is accomplished by programming the Sequence Controller using the panel labeled Test Sequences. Data may be transmitted in a single cycle, looped, or run continuously. Test sequences may be programmed to run in an automated mode, testing input signals or the contents of input data to start or stop sequences.

4.2.2.6.1 Define the Test Sequences

Test sequences are named by right clicking on Test Sequences in the menu directory. Test sequences can be referenced by any name the user wishes. Multiple test sequences can be defined to the limit of

the test sequence memory. Start the process by naming a new test sequence and entering an estimated number of steps needed to perform the sequence required in the “Sequence Size:” control. Steps can be inserted or deleted during step entry. The Words Available window will keep track of the memory space. After naming a Test Sequence, the step list panel for the Test Sequences will appear. The Edit Step button provides access to the programming panel to perform functions from simple outputs to more complex interactive functions involving conditional and unconditional, jumping, sub-routines, looping, etc.



VXI plug&play functions for defining a test sequence:

ta2108_txDefineSequence

4.2.2.6.2 Programming Sequence Steps

Test sequence programming is performed by highlighting the Step to be programmed and then clicking on the Edit Step button. The programming panel will appear. There are six command types that can be programmed.

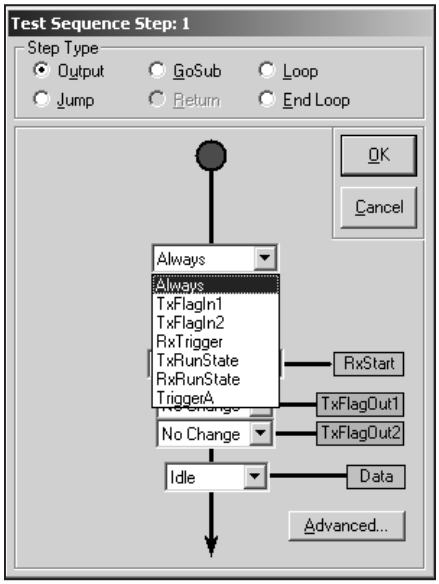
VXI plug&play functions for editing a test sequence step:

ta2108_txSelectSequenceStep

ta2108_txDeleteSequenceStep

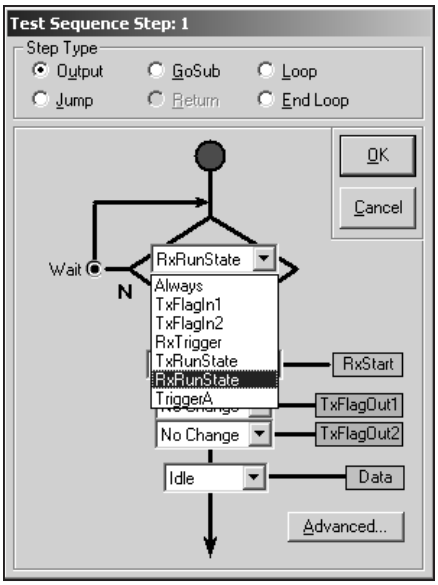
4.2.2.6.2.1 “Output” Sequence Step

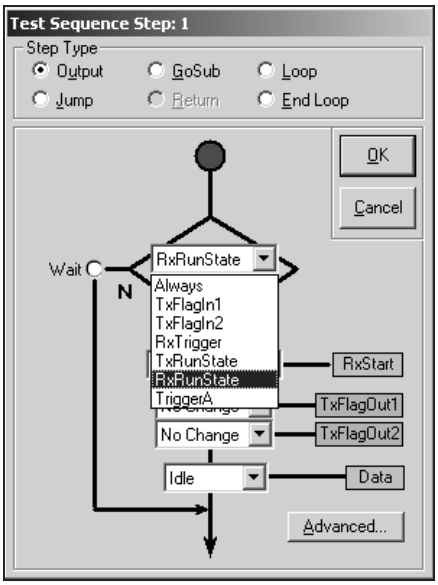
This command allows the user to output an RxStart command, TxFlagOut level, CMT or any combination of the three. The output can be unconditional “Always” or triggered by an event.



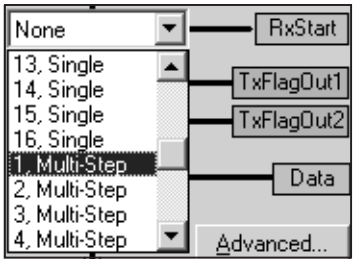
Events may be:

triggers from the UUT, a trigger or run state from an adjacent 2108RX, run state of the 2108TX, or a Trigger from the VXI bus. The output sequence step may be programmed to “Wait” for an event to be true before proceeding or if false skip this step and proceed to the next step.

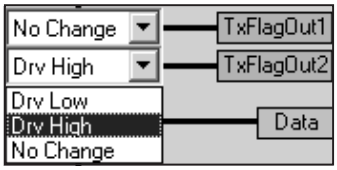




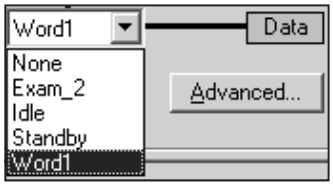
The RxStart selection window is used to arm an adjacent 2108RX trigger sequence as a single or multi step operation.



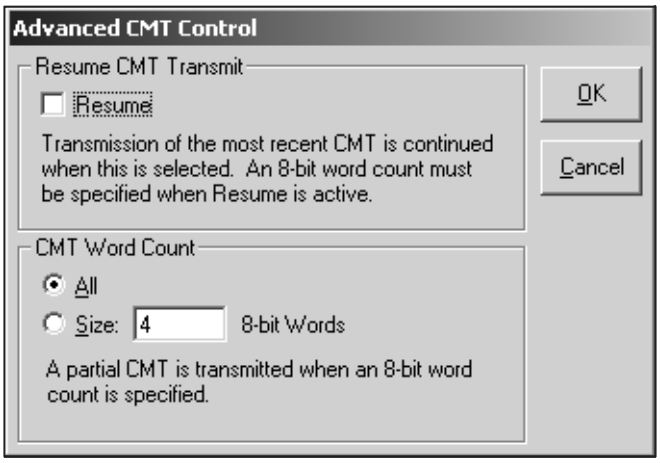
The second and third windows set the state of the transmit flag out signals. The signals may be individually programmed to drive high or low prior to the CMT Data being output.



The last window in this panel is the CMT selection window for the CMT to be output.



The “**Advanced...**” command allows the user to output a subset of the specified CMT or resume the previous CMT.

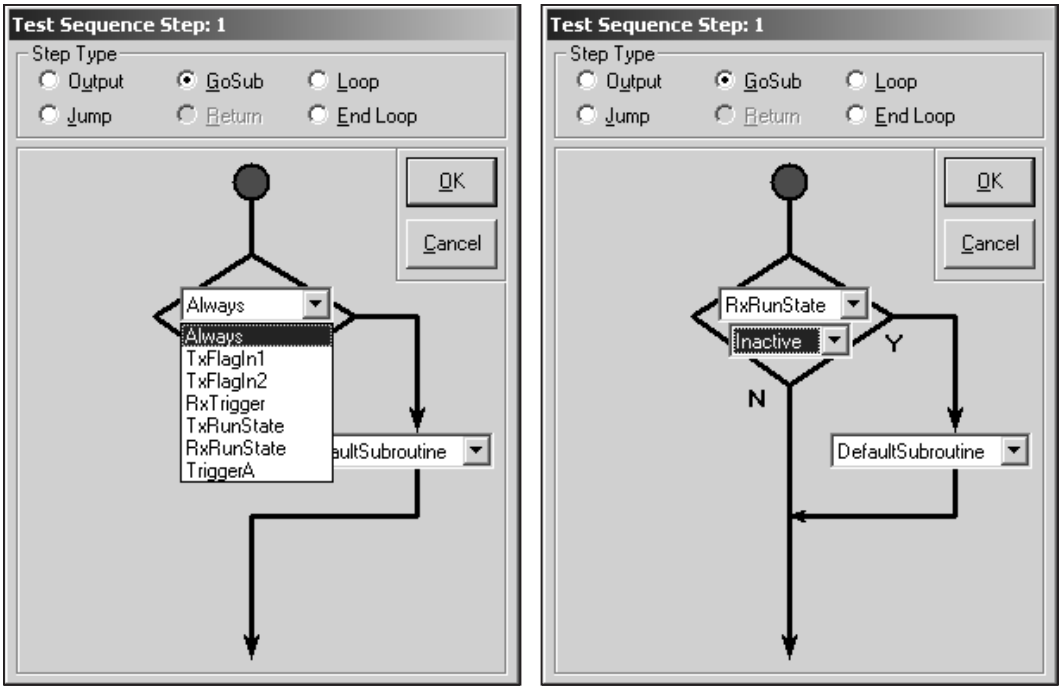


VXI plug&play functions for inserting a test sequence output step:

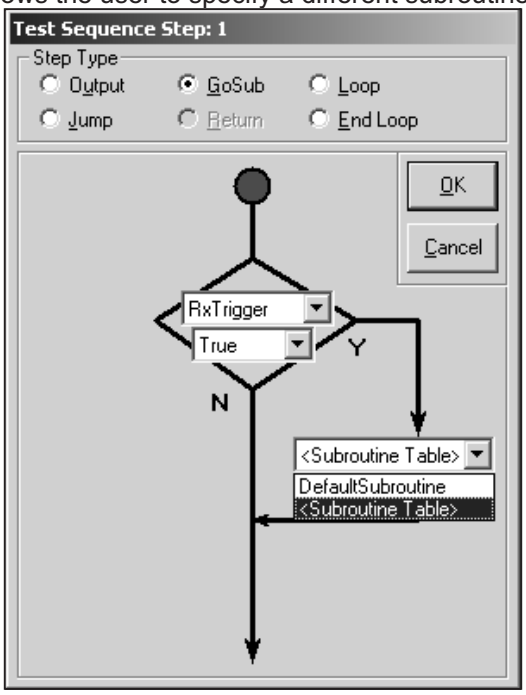
ta2108_txInsertSequenceOutputStep

4.2.2.6.2.2 “GoSub” Sequence Step

This command allows the user to call a test subroutine. Test Subroutines have an implied Return after execution. The selected subroutine may always be executed or conditionally executed. The selected subroutine must be defined and stored under the Test Subroutines section of the menu directory.



The “Subroutine Table” selections becomes active if the condition is set to “**RxTrigger**” and “**True**”. The RxTrigger Subroutine Table allows the user to specify a different subroutine for each trigger.

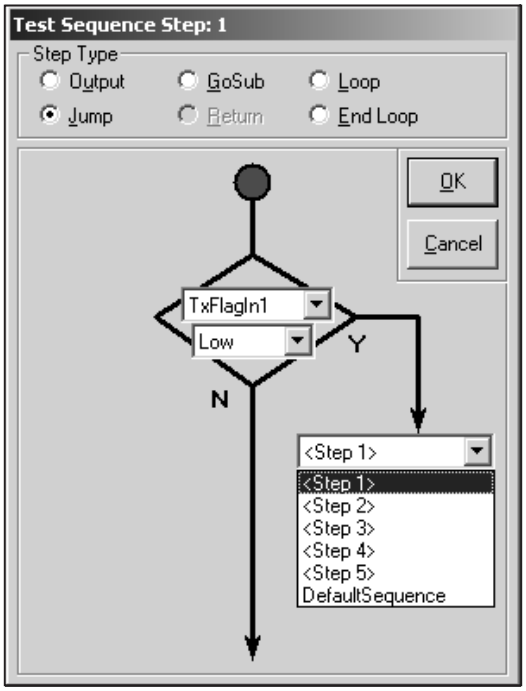


VXI plug&play functions for inserting a test sequence gosub step:

ta2108_txInsertSequenceGosubStep

4.2.2.6.2.3 “Jump” Sequence Step

This command allows the user to program a Jump to a specific step in the current test sequence or to the beginning of another sequence. The jump may “**Always**” be executed or conditionally executed.

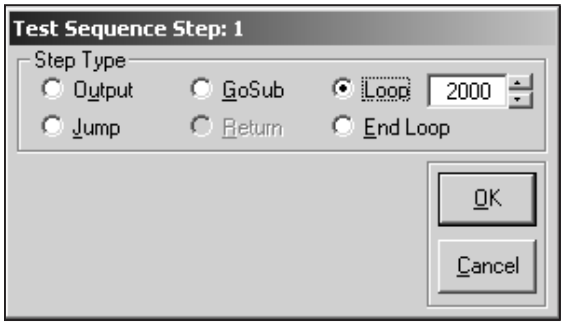


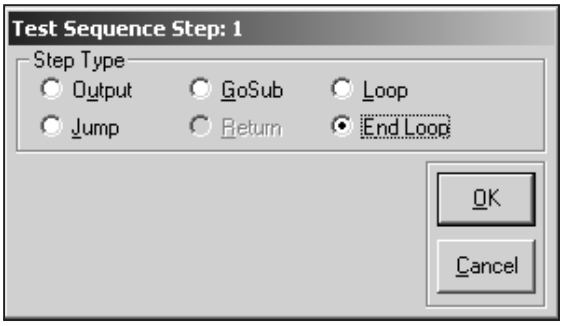
VXI plug&play functions for inserting a test sequence jump step:

ta2108_txInsertSequenceJumpStep

4.2.2.6.2.4 "Loop"/"End Loop" Sequence Steps

These commands allow the user to program loops from 1 to 32k times for all sequences between the Loop command and the End Loop command. In addition, Loops may be nested 3 levels deep.





VXI plug&play functions for inserting a test sequence loop/end loop step:

ta2108_txInsertSequenceLoopStep

ta2108_txInsertSequenceEndLoopStep

4.2.2.6.3 "Return" Sequence Step

This command is enabled only in test subroutines. Subroutines can be nested up to 16 levels. The return command exits the current level in the subroutine stack.

VXI plug&play functions for inserting a test sequence loop/end loop step:

ta2108_txInsertSequenceReturnStep

4.2.2.6.4 Sequence Step List

Once a Test Sequence is completed, the Step List provides the user with a complete listing of the program for documentation in an easy-to-read and understand format. There is also an entry area for descriptions.

Sequence size: 3 Words available: 939 Edit Step

Step	Command	Description
1	Flg(LH)	Set flag1 low and flag2 high
2	If InFlag1 = Low Then RxStart(1, Single): Flg(NN)	If input flag 1 low arm record step 1
3	Wait Until RxRunState = Active Then Xmit(Word1)	Wait until receiver finished and transmit "Word1"

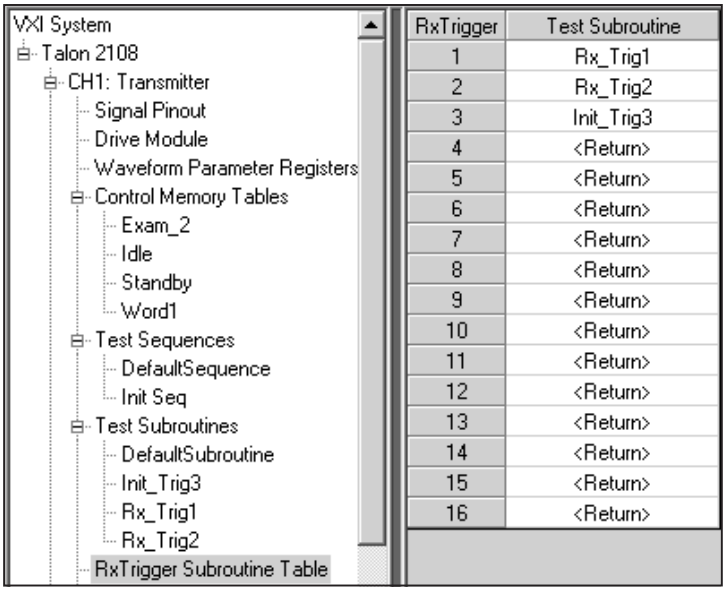
4.2.2.6.5 Test Subroutines

Test Subroutines are test sequences programmed as subroutines to be called from the main test sequences. This feature saves test sequence steps. Test Subroutines are named and programmed the same as Test Sequences except the Return command is activated.

4.2.2.6.6 RxTrigger Subroutine Table

RXTRIGGER Subroutine Table is a subset of the Test Subroutines. A 2108TX subroutine may be triggered by one of the 16 2108RX triggers. 2108RX triggers are set when the 2108RX receives a

specific UUT qualifier signal or data pattern. The unique trigger number is passed to the 2108TX and may be used to initiate an RXTRIGGER Subroutine Table output. This function is often used to perform a command/response operation.



RxTrigger	Test Subroutine
1	Rx_Trig1
2	Rx_Trig2
3	Init_Trig3
4	<Return>
5	<Return>
6	<Return>
7	<Return>
8	<Return>
9	<Return>
10	<Return>
11	<Return>
12	<Return>
13	<Return>
14	<Return>
15	<Return>
16	<Return>

VXI plug&play functions for inserting a test sequence loop/end loop step:

ta2108_txSetRxTriggerSubroutine

4.2.2.6.7 Test Sequence Examples

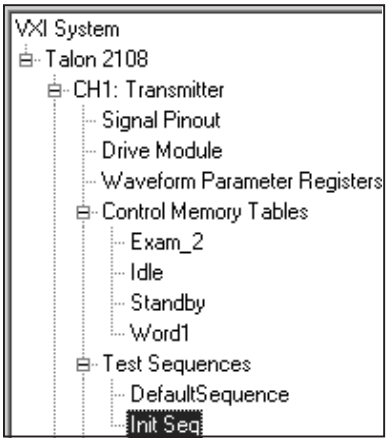
The following sections illustrates two test sequence examples.

4.2.2.6.7.1 Example 1: Transmitting a CMT When Triggered

In this example the user needs to accomplish something very simple, wait for an input trigger from the UUT prior to transmitting a CMT. The example test sequence will be named "**Init Seq**". The sequence will output a CMT named "**Word1**" when triggered by the rising edge of "**TxFlagIn1**". Programming steps are:

a) Right click on "Test Sequences" in the menu directory, select "Add..." and then "Test Sequence"

b) Enter a name for the test sequence, ("Init Seq" for this example)



c) Leave 1 as the Sequence Size

d) Click on Edit Step

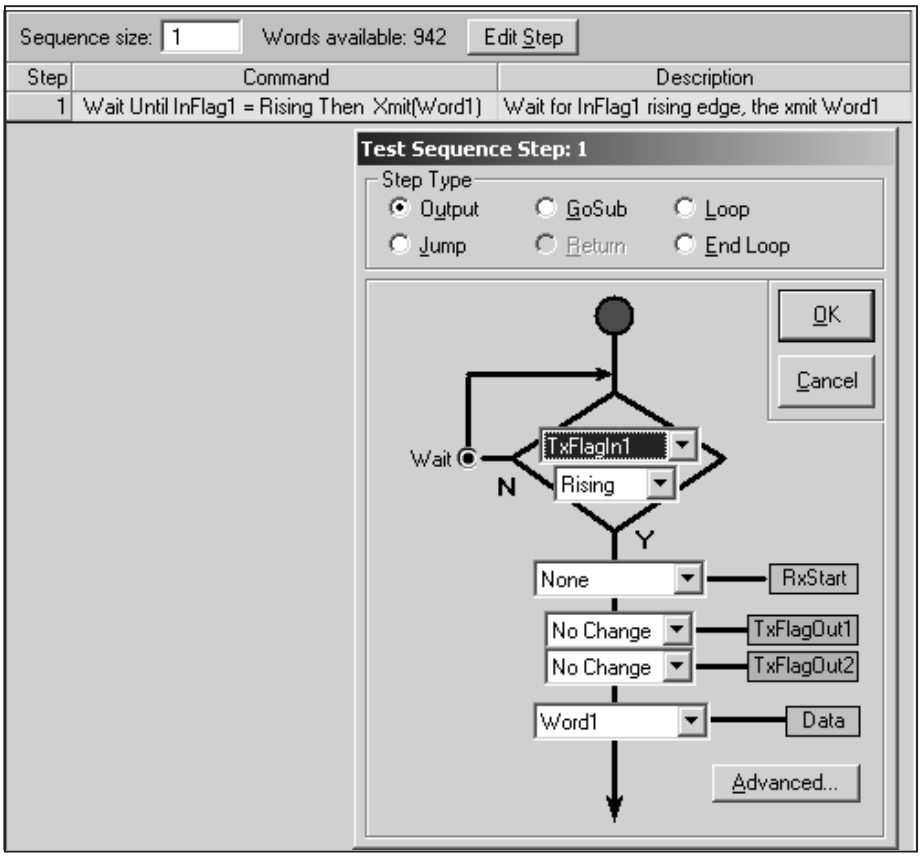
e) Select Output as the Step Type

f) Select TxFlagIn1 in the decision tree

g) Select Rising as the test

h) Click on Wait to set the 2108TX to wait for the rising edge

i) In the Data pull down window select “Word1” as the CMT to be transmitted

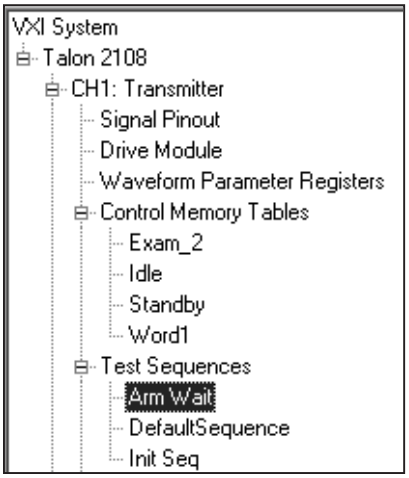


4.2.2.6.7.2 Example 2: Arm Receiver and Transmit a CMT When Receiver Triggered

In this example the user needs to arm the 2108 receiver, wait for the receiver to trigger, and then transmit a CMT. The example test sequence will be named, “**Arm Wait**” and it is to output CMT “**Exam_2**” 10 times when the receiver captures a valid trigger pattern. Programming steps are:

a) Right click on “Test Sequences” in the menu directory, select “Add...” and then “Test Sequence”

b) Enter a name for the sequence, (”Arm Wait” for this example)



c) Enter 5 as the Sequence Size
d) Edit step 1 and select “Output” as the Step Type
e) Select “Always” in the decision tree
f) Select “1,Single” for RxStart
g) Select “No Change” for TxFlagout1 and TxFlagOut2
h) Select “None” for Data
i) Edit step 2 and select Output as the Step Type
j) Select “RxTrigger”, “True”, “Wait” in the decision tree
k) Select “None” for RxStart
l) Select “No Change” for TxFlagout1 and TxFlagOut2
m) Select “None” for Data.
n) Edit step 3 and select Loop as the Step Type and enter 10
o) Edit step 4 and select Output as the Step Type
p) Select “Always” in the decision tree
q) Select “None” for RxStart
r) Select “No Change” for TxFlagout1 and TxFlagOut2
s) Select “Got Data” for Data
t) Edit step 5 and select “End Loop” as the Step Type

Sequence size: 5 Words available: 933 Edit Step

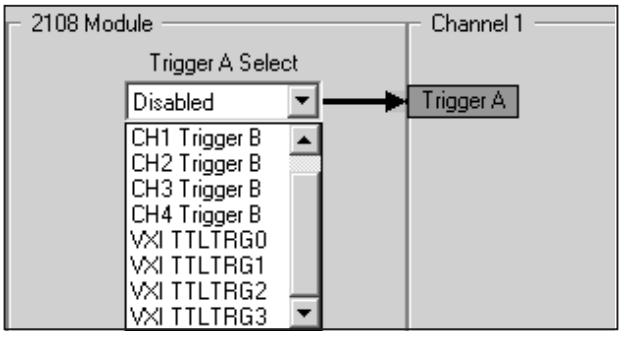
Step	Command	Description
1	RxStart(1, Single): Flg(NN)	Arm step 1 of receiver.
2	Wait Until RxTrigger = True Then Flg(NN)	Wait until receiver triggers.
3	Loop: 10	Set loop to 10.
4	Xmit(Exam_2)	Transmit "Exam_2" CMT.
5	End Loop	Repeat 10 times.

4.2.2.7 VXI Triggers & Interrupts

VXI Triggers & Interrupts menu directory provides means for mapping input and output triggers to/from the 2108TX.

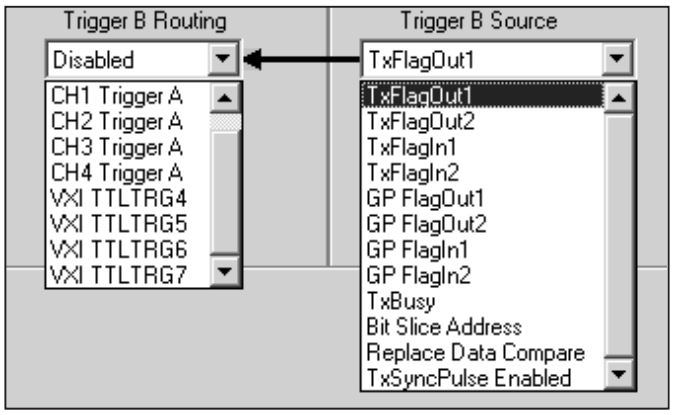
4.2.2.7.1 VXI Triggers

The VXI triggers, TTLTRG0, TTLTRG1, TTLTRG2, or TTLTRG4 may be mapped to the 2108TX as the source for TriggerA. In addition, Trigger B from any of the channel slots may be routed to TriggerA as the source.



TriggerB may be sourced from one of many inputs (transmitter flags in or out, GP flags, etc.) and routed

to one of the VXI triggers, TTLTRG4, TTLTRG5, TTLTRG6 or TTLTRG7. In addition, TriggerB may be routed to TriggerA of Chan1, Chan2, Chan3 or Chan4.



VXI plug&play functions for setting the 2108 triggers:

ta2108_txSetTriggerInput

ta2108_txSetTriggerOutput

4.2.2.7.2 VXI Interrupts

There are 24 functions in the Model 2108 which could generate interrupts for the VXI controller. This window allows any one of the 24 to be set as an interrupt. The interrupt service loop must be written in the test program and enabled and disabled as needed. Select an interrupt by using the pull down list and clicking on the function which is to generate the interrupt.

4.2.3 Configure the 2108 Receiver

The 2108RX was developed to receive and record serial data at rates to 200Mbps. The architecture of the 2108RX is best described as a 200MHz logic analyzer dedicated to serial applications.

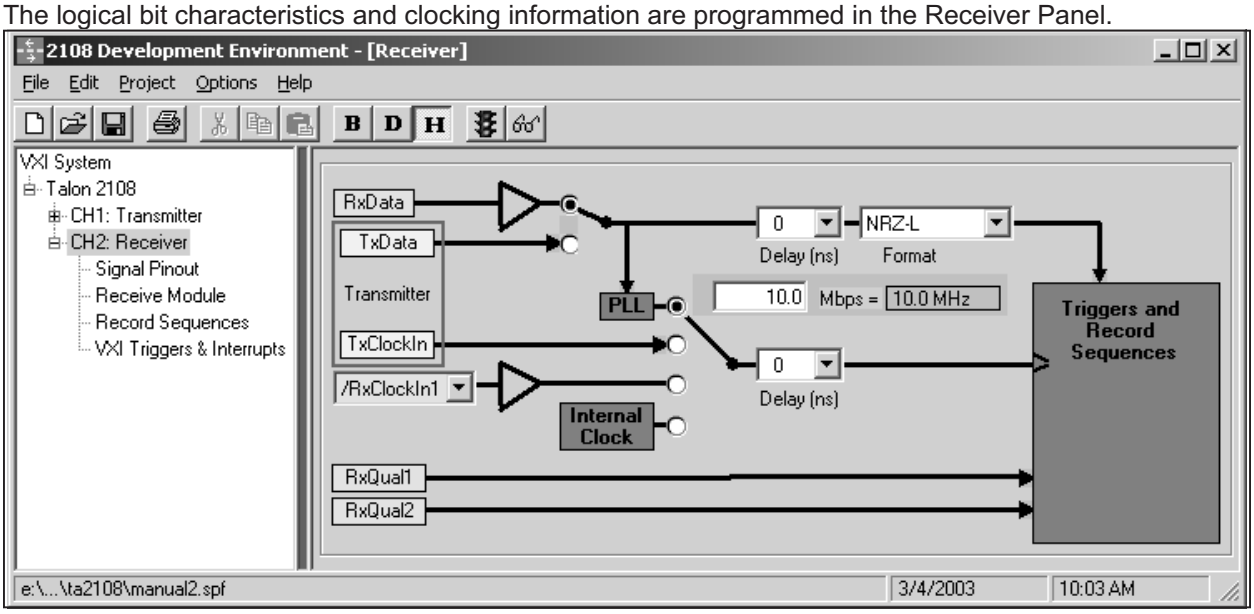
A total of 8Mb of data can be recorded (actually, there are two 8Mb memories, one storing “GOOD 1” and the other storing “GOOD 0” data). Alternately, the 2108RX can operate in a continuous recording mode where 4Mb are actively recorded while the previous 4Mb of recorded data are transferred to a mass storage media. Seamless recording is achieved by ping-ponging between the two 4Mb memory banks.

The operation of the 2108RX module requires the user to define the characteristics of the serial interface to be emulated. This includes the initiation of data recording as well as the bit characteristics, clock speed and other features of the serial interface to be emulated.

The characteristics of the serial bit include the logical and physical characteristics as well as the clocking of the data.

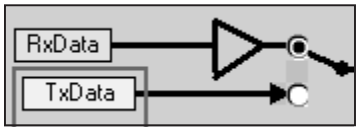
4.2.3.1 Logical Bit Settings

The logical bit characteristics and clocking information are programmed in the Receiver Panel.



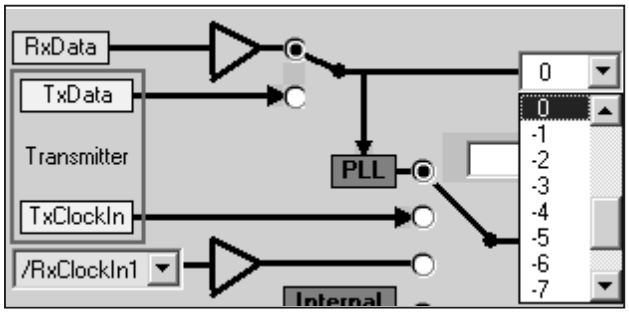
In the Receiver panel the user programs:

Data Source The data source selection allows the user to select either the front panel input(s) "**RxData**" or the adjacent transmitter output "**TxData**". If the "**TxData**" source is selected, then the adjacent transmitter channel must have the "**Bi-Directional**"

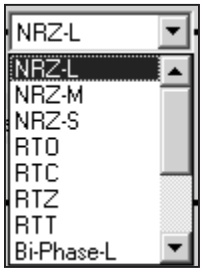


mode set in the "**Drive Module**" panel.

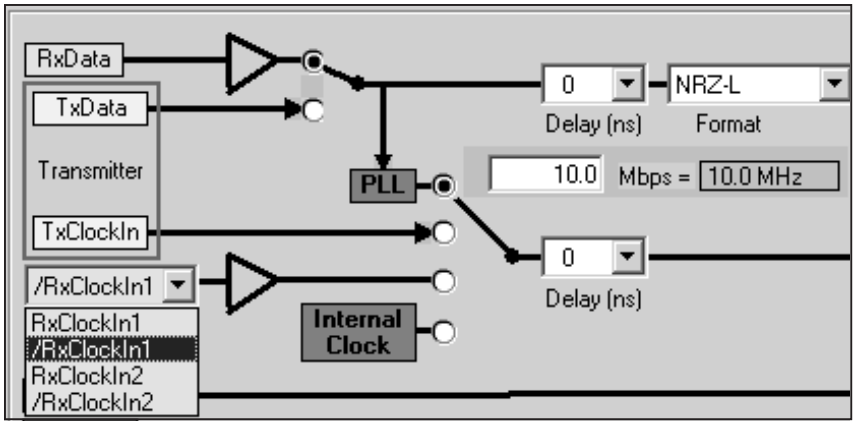
Data Delay A Delay in 1ns increments from +/- 10ns may be added to data paths to correct for synchronization problems due to cables, etc.



Data Format The receiver must know the type of data format in order to properly record the incoming data. The data format is selected using the pull-down menu.

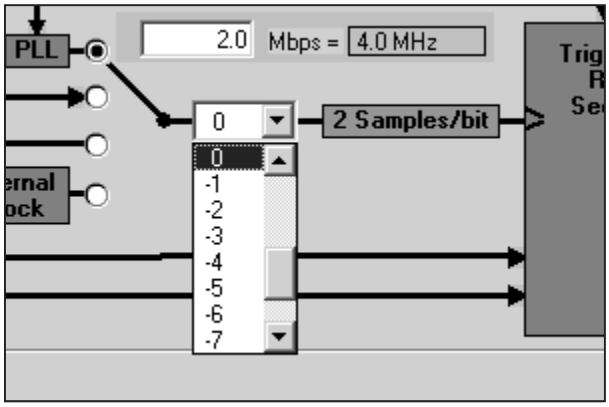


Clock Source The clock used to capture data can be set to any of the following:



Data	The clock is derived from the selected data source. The bit rate must be specified for the clock recovery logic to work.
TxClockIn	The selected external clock from the adjacent transmitter channel.
RxClockIn1	Front panel clock one (single ended ECL), inverted or non-inverted.
RxClockIn2	Front panel clock two (variable voltage), inverted or non-inverted.
Internal Generator	Internal clock generator.

Clock Delay A Delay in 1ns increments from +/- 10ns may be added to the clock paths to correct for synchronization problems due to cables, etc.



VXI plug&play functions for setting the 2108 receiver logical bit settings:

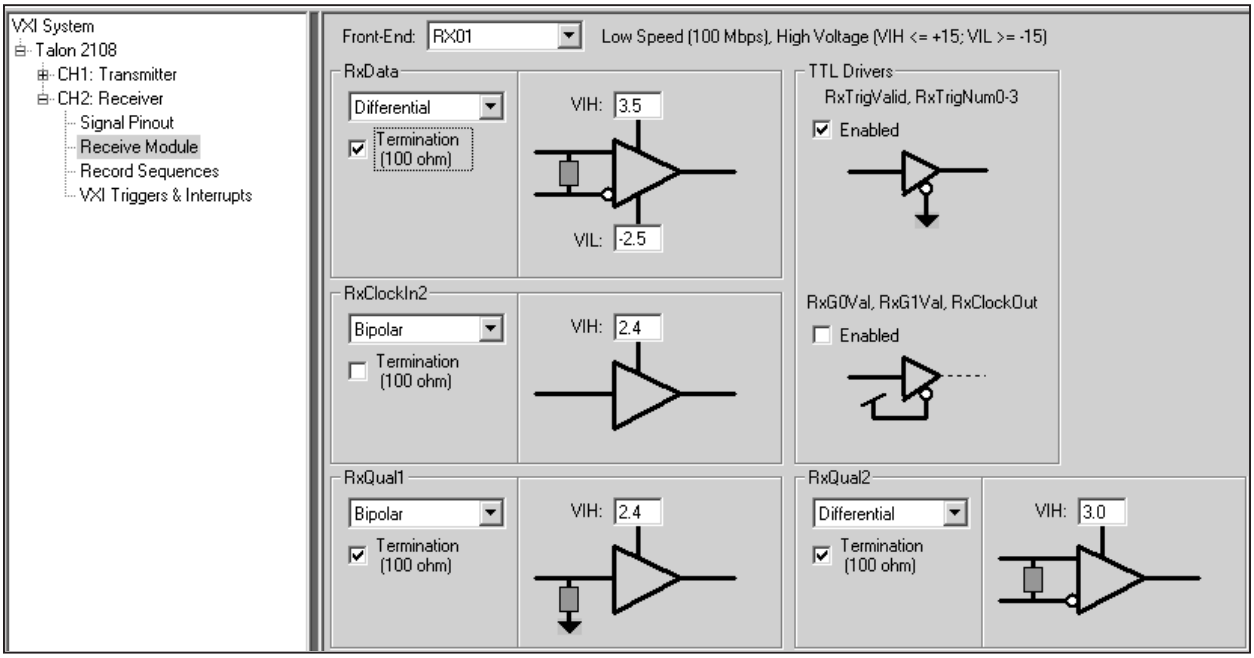
ta2108_rxSetSignalParameters

ta2108_rxSetSignalDelay

ta2108_rxSetFrontEndControl

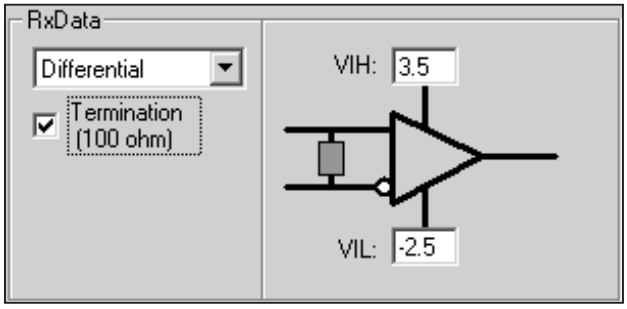
4.2.3.2 Electrical Characteristics

The physical bit characteristics, data voltage levels, signal types, etc., for the record data are programmed in the Receive Module panel.

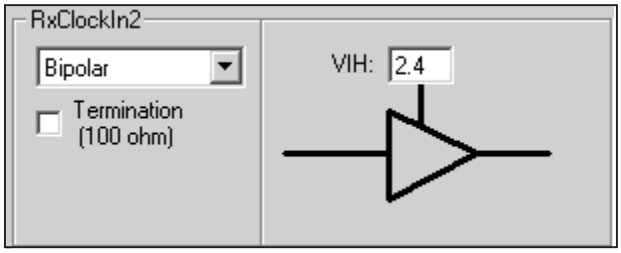


The signals to be programmed are:

RxData The selections to be programmed for the receive data are the type, Differential or Bi-polar, Termination or none, and the voltage levels VIH and VIL.

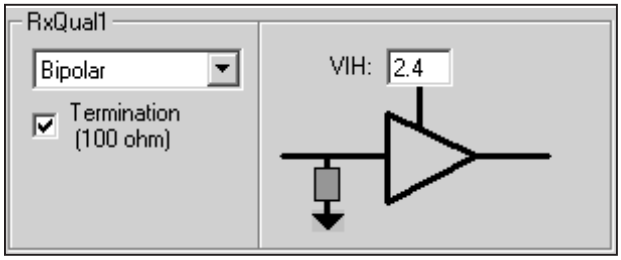


RxClockIn2 The receiver external clock input, **RxClockIn2**, is programmable. The signal type, termination signal and the “Good” level is programmed in this panel.



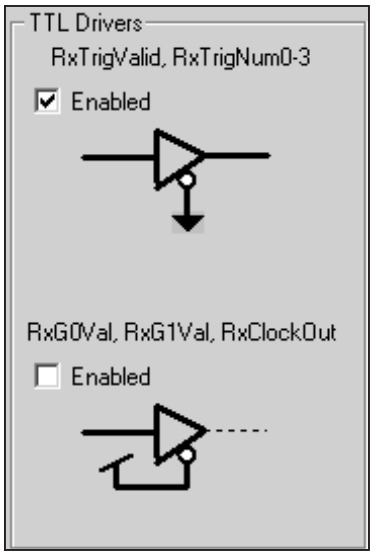
RxQual1/RxQual2 The 2108RX is capable of being triggered by an external trigger source.

RxQual1 and **RxQual2** are the two qualifier signals and both are programmable. The signal type, termination and “Good” signal level may be programmed.



TTL Drivers

The RX01 and RX02 drive the user selectable signals explained in the Signal Pinouts section above with TTL drivers. The drivers may be enabled or not depending on whether the user is using them in the project. This panel allows the user to set the state of the drivers.



VXI plug&play functions for setting the 2108 receiver electrical characteristics:

ta2108_rxSetFrontEndControl

ta2108_rxSetFrontEndInput

ta2108_rxSetReceiverReference

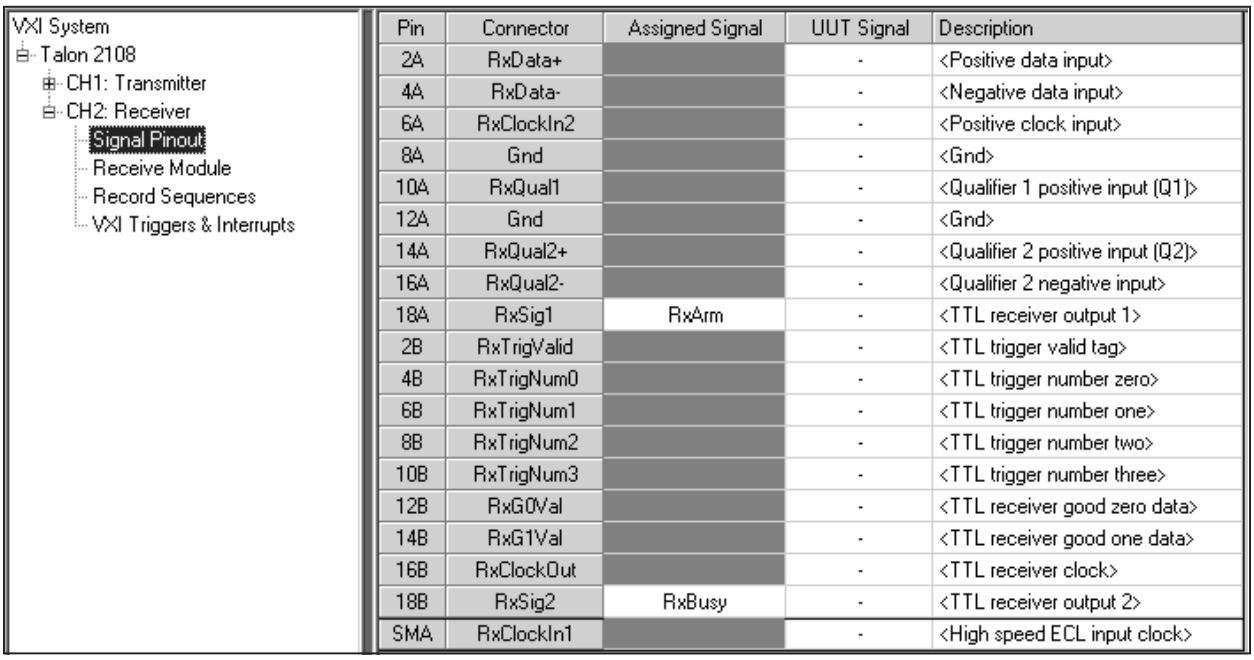
4.2.3.3 Routing the 2108rx signal pinouts

The 2108RX provides two pins which are user assignable through the Signal Pinot panel. These pins are 18A and 18B. The signals which may be assigned to the pins are :

RxArm	(RxSig1 default) A high level indicates that a record sequence is running and waiting for trigger.
RxBusy	(RxSig2 default) A high level indicates that a record sequence is running.
RxWait	A high level indicates the record sequence is waiting for TxAck from adjacent transmitter.
TxAck	Acknowledge signal from adjacent transmitter.
LostClk	A high level indicates a clock recovery error.
HFCR-ERR	A high level indicates a High Frequency Clock Recovery error.
MemABsy	A high level indicates that record memory bank A is busy.
LFCR-DEV	A high level indicates a Low Frequency Clock Recovery deviation (too slow or fast).

The remainder of the signals are not assignable by the user on the standard I/O modules, (RX01 and

RX02). The Signal Pinot panel also allows the user to enter UUT pin names and descriptions for each pin for documentation purposes.



Pin	Connector	Assigned Signal	UUT Signal	Description
2A	RxData+		-	<Positive data input>
4A	RxData-		-	<Negative data input>
6A	RxClockIn2		-	<Positive clock input>
8A	Gnd		-	<Gnd>
10A	RxQual1		-	<Qualifier 1 positive input (Q1)>
12A	Gnd		-	<Gnd>
14A	RxQual2+		-	<Qualifier 2 positive input (Q2)>
16A	RxQual2-		-	<Qualifier 2 negative input>
18A	RxSig1	RxArm	-	<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	-	<TTL receiver output 2>
SMA	RxClockIn1		-	<High speed ECL input clock>

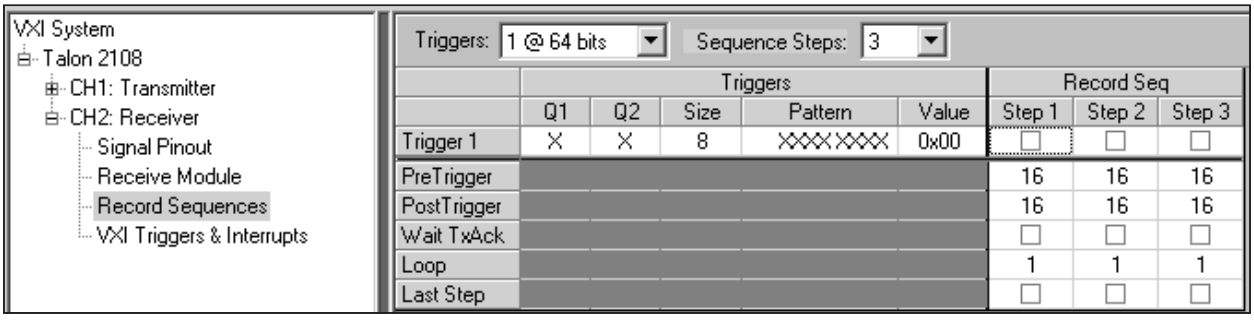
VXI plug&play functions for setting the 2108 receiver signal Pinouts:

ta2108_rxSetUserSignal

4.2.3.4 Defining Record Sequences

The Record Sequences panel is used to define the trigger conditions to initiate the recording of data by the 2108RX. The conditions may be just a single signal going high or more complex scenarios involving data patterns.

The 2108RX Record Sequencer allows a multitude of trigger combinations to be defined. A record sequence consists of two basic elements, the "**Trigger Condition(s)**" and the "**Record Sequence Step(s)**".



4.2.3.4.1 Defining Trigger Conditions

There are 16 thirty-two bit trigger conditions, named **Trigger 1** through **Trigger 16**. These triggers can be configured via software as 8 sixty-four bit conditions.

Each trigger condition is comprised of the following:

Q1/Q2 Expected state of the **RxQual1/RxQual2"** front panel signals.

"**X**" Don't care.

"**L**" **RxQual1** must be low for a valid trigger.

"**H**" **RxQual2** must be high for a valid trigger.

Size Number of bits in the trigger pattern definition (4 to 64).

Pattern The pattern is defined by individual bit state characters. 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.

VXI plug&play functions for setting the 2108 receiver trigger conditions:

ta2108_rxSetTrigger

4.2.3.4.2 Define Record Sequence Steps

There are 16 record sequence steps. Each record sequence step is comprised of the following settings:

Trigger List 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.

Pre Trigger Bits The Pre trigger fields define the number of samples that will be recorded before the trigger condition. Valid range is from 16 to 32752.

Post Trigger Bits The Post trigger fields define the number of samples that will be recorded after the trigger condition. The Post trigger value can be set to negative one (-1) in order to sample continuously. Valid range is from 16 to 8388608.

Wait TxAck Flag Pauses the record sequence until a handshake signal (Transmit Acknowledge) is generated by the adjacent transmit channel.

Loop Count 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). Valid range is from 1 to 65535.

Last Step Flag The Last Step field is used when a series of record sequence steps are to be applied. The user 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.

VXI plug&play functions for setting the 2108 receiver record sequence steps:

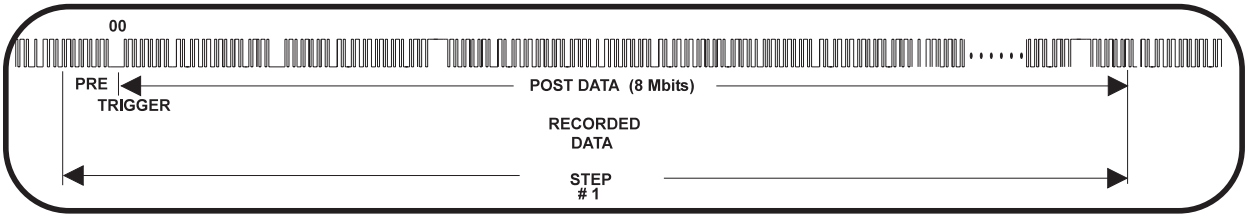
ta2108_rxSetRecordSequence

4.2.3.4.3 Record Sequence Examples

The following sections illustrate several record sequence examples.

4.2.3.4.3.1 Example #1: Trigger and Record

Assume it is desired to trigger on the first occurrence of the serial hexadecimal character h00.



To achieve this, **Trigger 1** would be set to h00. A single sequence step would be entered. **Step 1**, **Trigger 1** would then be enabled (shown with a check mark). The **Pre Trigger** bit length is set to 64 bits, the **Post Trigger** bit length is set to 8,000,000 bits, the **Loop** count set to 1 and **Last Step** enabled.

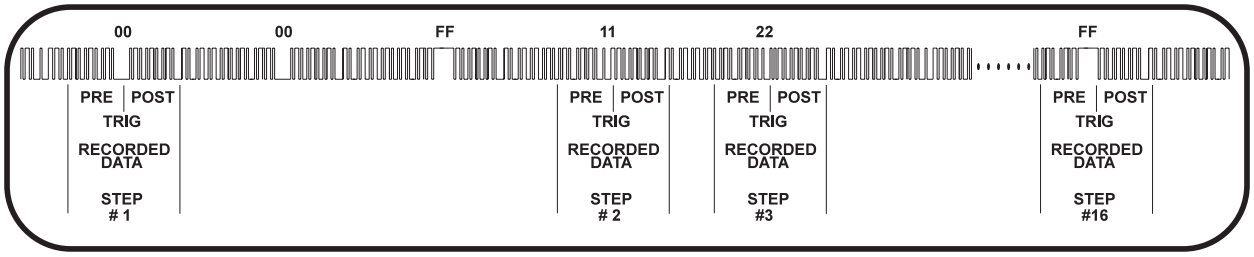
Triggers: 1 @ 64 bits Sequence Steps: 1

	Triggers				Record Seq
	Q2	Size	Pattern	Value	Step 1
Trigger 1	X	8	XXXX XXXX	0000 0000b	☑
PreTrigger					64
PostTrigger					8000000
Wait TxAck					☐
Loop					1
Last Step					☑

Once the serial data value of h00 is detected, the 2108RX will save the previous 64 data bits prior to the trigger and then record an additional 8 megabits of data after the trigger.

4.2.3.4.3.2 Example #2: Search for a Sequence of Data Values

Assume it is desired to record the data surrounding a sequential set of characters. In particular, assume we first want to record 64 bits (32 PRE DATA bits and 32 POST DATA bits) when the hex character h00 is detected. We then want to wait for the hex character h11, then h22 and so forth.



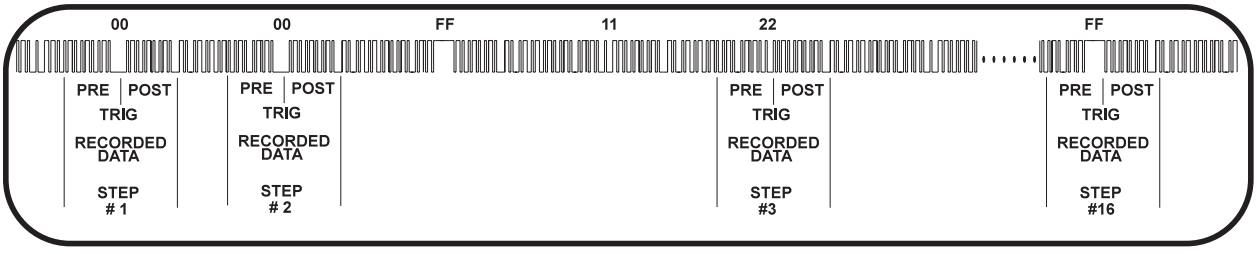
Step 1 would be enabled to search for the hex characters h00 with the **Loop** set to the value 1. After the trigger occurs, the sequencer would go to **Step 2**, which in turn is enabled to search for the value h11. The process will continue until **Step 16** is executed.

Triggers: 16 @ 32 bits Sequence Steps: 16

	Triggers				Record Seq								
	Q2	Size	Pattern	Value	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 16
Trigger 1	X	8	0000 0000	0x00	☑	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 2	X	8	0001 0001	0x11	☐	☑	☐	☐	☐	☐	☐	☐	☐
Trigger 3	X	8	0010 0010	0x22	☐	☐	☑	☐	☐	☐	☐	☐	☐
Trigger 4	X	8	0011 0011	0x33	☐	☐	☐	☑	☐	☐	☐	☐	☐
Trigger 5	X	8	0100 0100	0x44	☐	☐	☐	☐	☑	☐	☐	☐	☐
Trigger 6	X	8	0101 0101	0x55	☐	☐	☐	☐	☐	☑	☐	☐	☐
Trigger 7	X	8	0110 0110	0x66	☐	☐	☐	☐	☐	☐	☑	☐	☐
Trigger 8	X	8	0111 0111	0x77	☐	☐	☐	☐	☐	☐	☐	☑	☐
Trigger 9	X	8	1000 1000	0x88	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 10	X	8	1001 1001	0x99	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 11	X	8	1010 1010	0xAA	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 12	X	8	1011 1011	0xBB	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 13	X	8	1100 1100	0xCC	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 14	X	8	1101 1101	0xDD	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 15	X	8	1110 1110	0xEE	☐	☐	☐	☐	☐	☐	☐	☐	☐
Trigger 16	X	8	1111 1111	0xFF	☐	☐	☐	☐	☐	☐	☐	☐	☑
PreTrigger					32	32	32	32	32	32	32	32	32
PostTrigger					32	32	32	32	32	32	32	32	32
Wait TxAck					☐	☐	☐	☐	☐	☐	☐	☐	☐
Loop					1	1	1	1	1	1	1	1	1
Last Step					☐	☐	☐	☐	☐	☐	☐	☐	☑

4.2.3.4.3.3 Example #3: Search for Multiple Data Values

If two or more triggers are enabled in a particular step, the 2108RX will search and trigger if any of the trigger values are detected.

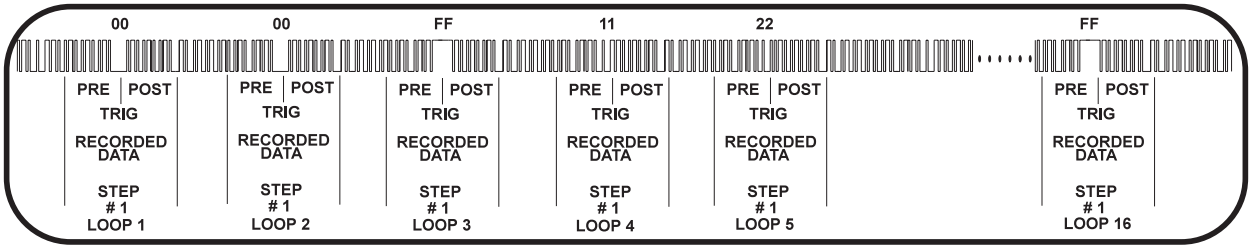


If **Trigger 1** is also enabled in **Step 2**, the sequence logic will now search for the occurrence of either h00 or h11.

Triggers: 16 @ 32 bits Sequence Steps: 16

	Triggers				Record Seq									
	Q2	Size	Pattern	Value	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	•••	Step 16
Trigger 1	X	8	0000 0000	0x00	☑	☑	☐	☐	☐	☐	☐	☐		☐
Trigger 2	X	8	0001 0001	0x11	☐	☑	☐	☐	☐	☐	☐	☐		☐
Trigger 3	X	8	0010 0010	0x22	☐	☐	☑	☐	☐	☐	☐	☐		☐
Trigger 4	X	8	0011 0011	0x33	☐	☐	☐	☑	☐	☐	☐	☐		☐
Trigger 5	X	8	0100 0100	0x44	☐	☐	☐	☐	☑	☐	☐	☐		☐
Trigger 6	X	8	0101 0101	0x55	☐	☐	☐	☐	☐	☑	☐	☐		☐
Trigger 7	X	8	0110 0110	0x66	☐	☐	☐	☐	☐	☐	☑	☐		☐
Trigger 8	X	8	0111 0111	0x77	☐	☐	☐	☐	☐	☐	☐	☑		☐
Trigger 9	X	8	1000 1000	0x88	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 10	X	8	1001 1001	0x99	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 11	X	8	1010 1010	0xAA	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 12	X	8	1011 1011	0xBB	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 13	X	8	1100 1100	0xCC	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 14	X	8	1101 1101	0xDD	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 15	X	8	1110 1110	0xEE	☐	☐	☐	☐	☐	☐	☐	☐		☐
Trigger 16	X	8	1111 1111	0xFF	☐	☐	☐	☐	☐	☐	☐	☐		☑
PreTrigger					32	32	32	32	32	32	32	32		32
PostTrigger					32	32	32	32	32	32	32	32		32
Wait TxAck					☐	☐	☐	☐	☐	☐	☐	☐		☐
Loop					1	1	1	1	1	1	1	1		1
Last Step					☐	☐	☐	☐	☐	☐	☐	☐		☑

The following figures depict the recorded data and sequence set up menu when all the triggers are enabled.

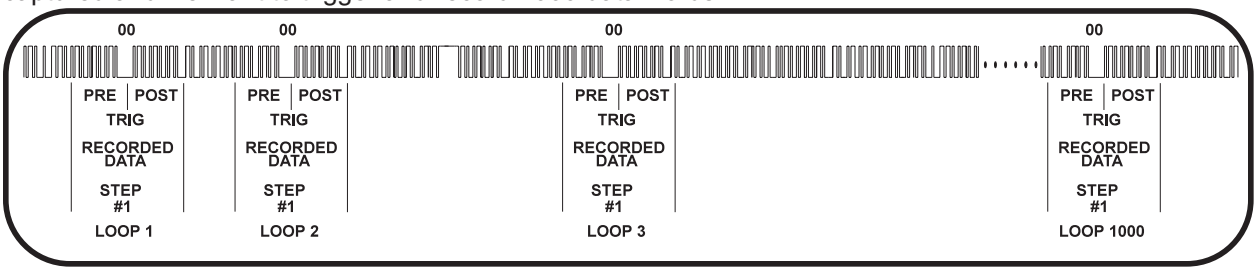


For **Step 1** note the **Loop** count is set to 16.

	Triggers					Record Se
	Q1	Q2	Size	Pattern	Value	Step 1
Trigger 1	X	X	8	0000 0000	0x00	☑
Trigger 2	X	X	8	0001 0001	0x11	☑
Trigger 3	X	X	8	0010 0010	0x22	☑
Trigger 4	X	X	8	0011 0011	0x33	☑
Trigger 5	X	X	8	0100 0100	0x44	☑
Trigger 6	X	X	8	0101 0101	0x55	☑
Trigger 7	X	X	8	0110 0110	0x66	☑
Trigger 8	X	X	8	0111 0111	0x77	☑
Trigger 9	X	X	8	1000 1000	0x88	☑
Trigger 10	X	X	8	1001 1001	0x99	☑
Trigger 11	X	X	8	1010 1010	0xAA	☑
Trigger 12	X	X	8	1011 1011	0xBB	☑
Trigger 13	X	X	8	1100 1100	0xCC	☑
Trigger 14	X	X	8	1101 1101	0xDD	☑
Trigger 15	X	X	8	1110 1110	0xEE	☑
Trigger 16	X	X	8	1111 1111	0xFF	☑
PreTrigger						32
PostTrigger						32
Wait TxAck						☐
Loop						16
Last Step						☑

4.2.3.4.3.4 Example #4: Record a Multiple Number of Words

Each sequence step incorporates a **Loop** count value. This enables the recording of multiple words, each with a common or unique trigger value. Assume the code h00 precedes each data word to be captured and we want to trigger and record 1000 data words.

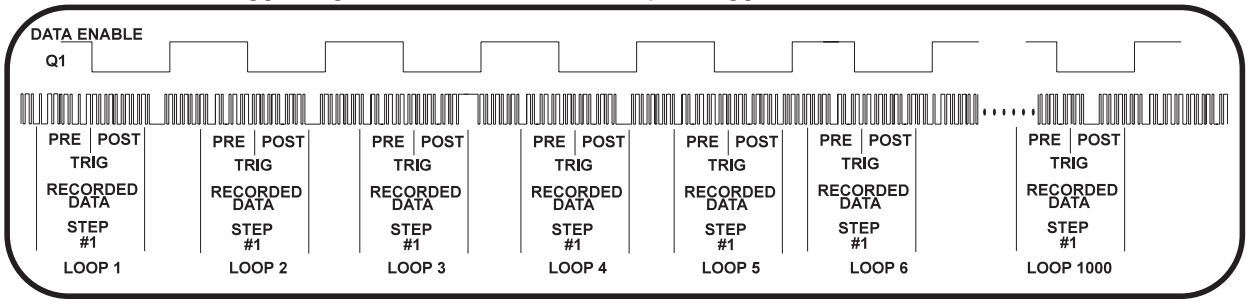


Triggers: 1 @ 64 bits Sequence Steps: 1

	Triggers					Record Se
	Q1	Q2	Size	Pattern	Value	Step 1
Trigger 1	X	X	8	0000 0000	0x00	☑
PreTrigger						32
PostTrigger						32
Wait TxAck						☐
Loop						1000
Last Step						☑

4.2.3.4.3.5 Example #5: Trigger on Qualifier

In addition to the 16 trigger registers, there are two qualifier signals, **Q1** and **Q2**, which can be logically combined with the trigger registers to form a more complex trigger event.



Assume an interface uses a separate signal, DATA ENABLE, to indicate that the serial data is valid. The sequence is set up to record 1000 words, one for each time the DATA ENABLE goes true (Active Low).

Triggers: 1 @ 64 bits Sequence Steps: 1

	Triggers					Record Se
	Q1	Q2	Size	Pattern	Value	Step 1
Trigger 1	L	X	8	XXXX XXXX	0x00	☑
PreTrigger						32
PostTrigger						32
Wait TxAck						☐
Loop						1000
Last Step						☑

The qualifiers can also be logically combined with the trigger registers.

The following figures depict the recording of data when the **Q1** is true and the data value of hFF is detected.

Triggers: 2 @ 64 bits Sequence Steps: 2

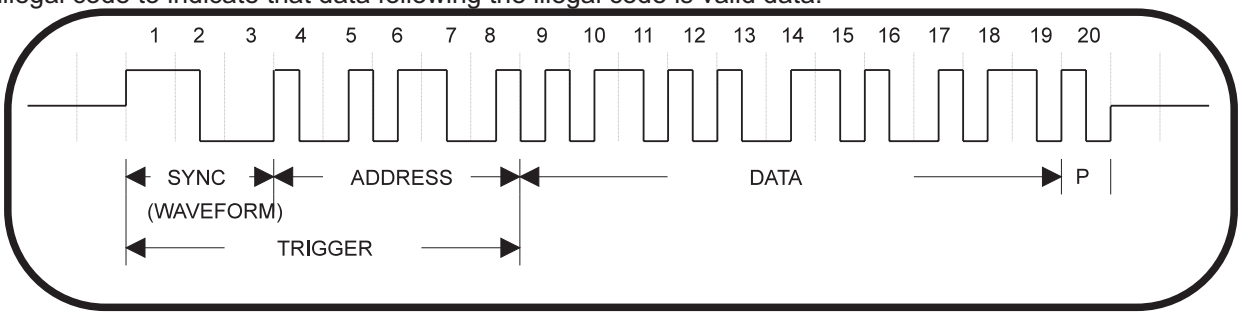
	Triggers					Record Seq	
	Q1	Q2	Size	Pattern	Value	Step 1	Step 2
Trigger 1	L	X	8	1111 1111	0xFF	☑	☐
Trigger 2	L	X	8	XXXX XXXX	0x00	☐	☑
PreTrigger						32	32
PostTrigger						32	32
Wait TxAck						☐	☐
Loop						1	1000
Last Step						☐	☑

After this occurrence, **Step 2** will record 1000 data words only when DATA ENABLE is true.

4.2.3.4.3.6 Example #6: Trigger on Waveform

Each data bit of a serial bus consists of a particular bit format; AMI, BiPhase, NRZ, etc. The 2108RX compensates for the format when the user enters his trigger data in a logical format.

Many serial interfaces use illegal data formats for the purpose of syncing the transmitter with the receiver such as the 1553 interface. A legal Manchester code does not allow for three consecutive ½ bit times of ones followed by three consecutive ½ bit times of zeros. The 1553 takes advantage of this by using this illegal code to indicate that data following the illegal code is valid data.



The trigger pattern can be specified as either a physical level (H, L, Z, X) or a logical level (1, 0, U).

Triggers: 1 @ 64 bits Sequence Steps: 1

	Triggers					Record Se
	Q1	Q2	Size	Pattern	Value	Step 1
Trigger 1	X	X	11	HHH LLL1 0010	0x712	☑
PreTrigger						16
PostTrigger						32
Wait TxAck						☐
Loop						1
Last Step						☑

4.2.3.5 VXI Triggers & Interrupts

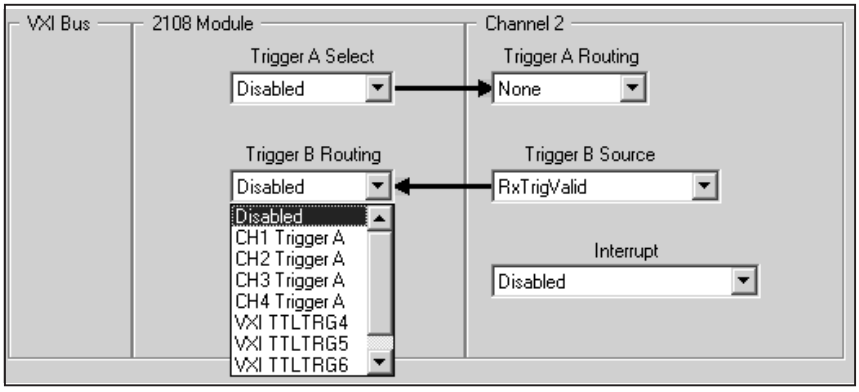
VXI Triggers & Interrupts menu directory provides for mapping of input and output triggers to/from the 2108RX.

4.2.3.5.1 VXI Triggers

The VXI triggers, TTLTRG0, TTLTRG1, TTLTRG2, or TTLTRG3 may be mapped to the 2108RX as the source for “Trigger A”. In addition, Trigger B from any of the four channel slots may be routed to TriggerA as the source.

TriggerB is sourced by the “RxTrigValid” flag and can be routed to one of the VXI triggers, TTLTRG4,

TTLTRG5, TTLTRG6 or TTLTRG7. In addition, from any of the TriggerB may be routed to TriggerA of Chan1, Chan2, Chan3 or Chan4.



VXI plug&play functions for setting the 2108 receiver triggers:

ta2108_rxSetTriggerInput

ta2108_rxSetTriggerOutput

4.3 Enable the Output Drivers.

The output drivers on the transmitter/receiver are enabled when any of the following functions are called:

ta2108_Execute

ta2108_Stop

ta2108_txEnable

ta2108_txRunCMT

ta2108_txRunSequence

ta2108_txStop

ta2108_rxEnable

The following functions disable the output drivers:

ta2108_Halt

ta2108_Rst

ta2108_txDisable

ta2108_txHalt

ta2108_reset

ta2108_rxDisable

4.4 Issue the EXECUTE commands.

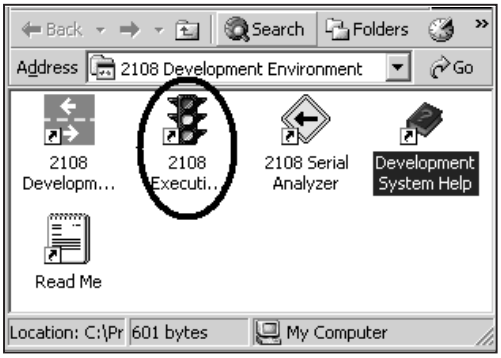
There are two methods of executing 2108 channels, interactive and programmatic. Interactive execution is useful for test development and debugging. Programmatic execution is handled via instrument driver functions.

There are two programs that are available for interactive execution, the Execution Manager and the Soft Front Panel. Both programs require that the Project Development Editor be used to configure the 2108 channels.

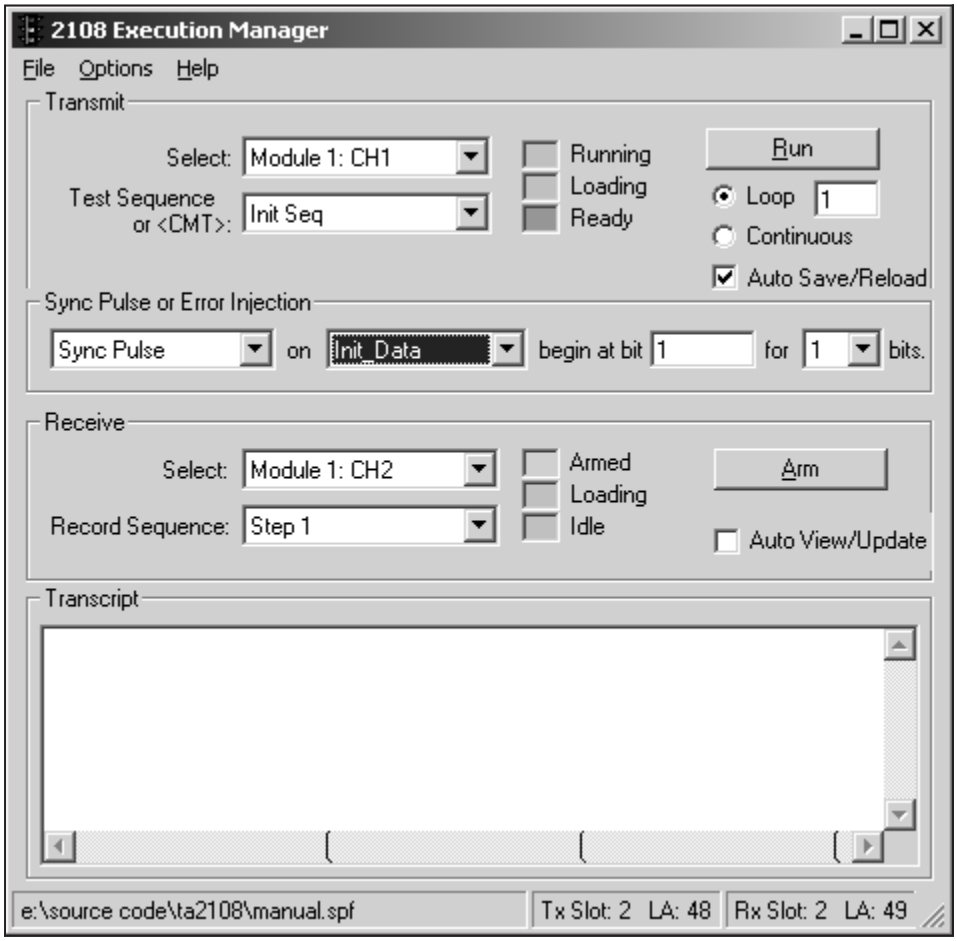
4.4.1 Execution Manager Panel

The Execution Manager is part of the Development system and allows the user to load and execute project editor files. The Execution Manager can be started from either the Project Editor by hitting the

Execution Manager icon on the menu bar or by clicking the Execution Manager program icon from the Development System folder.



The following figure illustrates the Execution Manager panel.



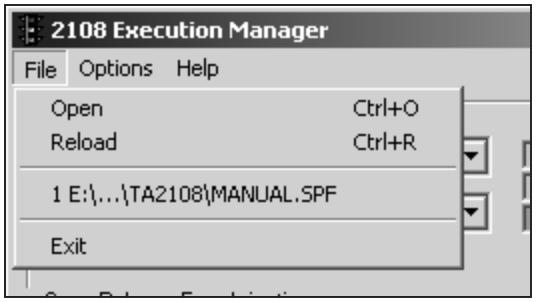
The 2108 Execution Manager is designed to simultaneously control one transmit and one receive channel. The 2108 Execution Manager window contains four control sections labeled “Transmit”, “Sync Pulse or Error Injection”, “Receive” and “Transcript”.

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.

4.4.1.1 Loading the Project File

If the Execution Manager is started from the Project Editor, the project is loaded automatically.

To load a new project file or reload the current project file select "**File**" on the menu bar.



4.4.1.2 Execution Manager Transmit

The Transmit section contains "Select" and "Test Sequence of <CMT>" 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 and 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.

VXI plug&play functions for executing the 2108 transmitter:

ta2108_Select

ta2108_Execute

ta2108_txRunSequence

ta2108_txRunCMT

4.4.1.3 Execution Manager Sync Pulse or Error Injection

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 CMT, 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.

VXI plug&play functions for setting the sync pulse or error injection:

ta2108_txSetSyncPulse

4.4.1.4 Execution Manager Receive

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 saved to disk and displayed by the Serial 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.

VXI plug&play functions for executing the 2108 transmitter:

ta2108_Select

ta2108_Execute

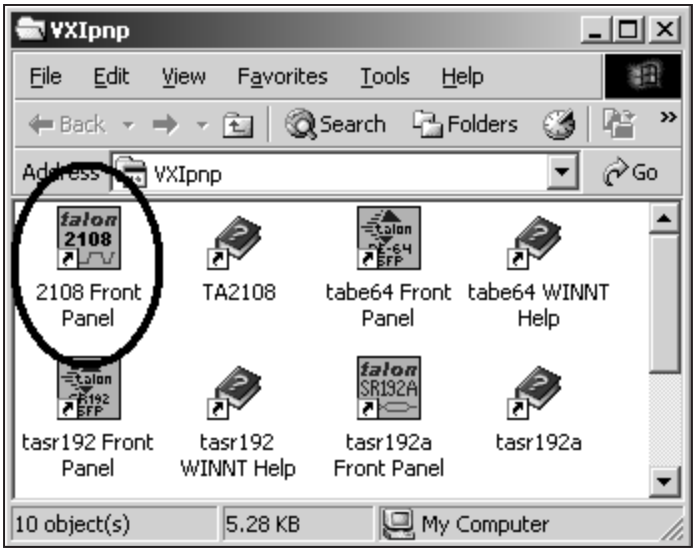
ta2108_rxArm

4.4.1.5 Execution Manager Transcript

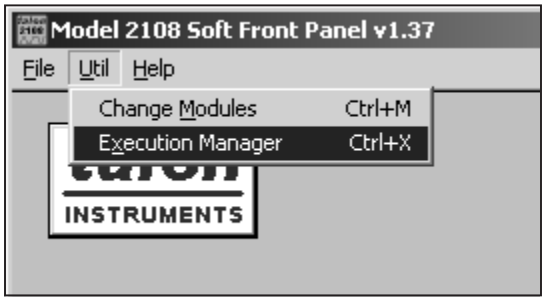
The Transcript window is a continuous journal of the test activity and test results. Its contents can be copied, pasted or cleared with the Options Menu. When the Transcript window buffer is full, a portion of the oldest data is deleted. During normal operation, the Transcript window logs start, end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

4.4.2 Soft Front Panel Execution Manager

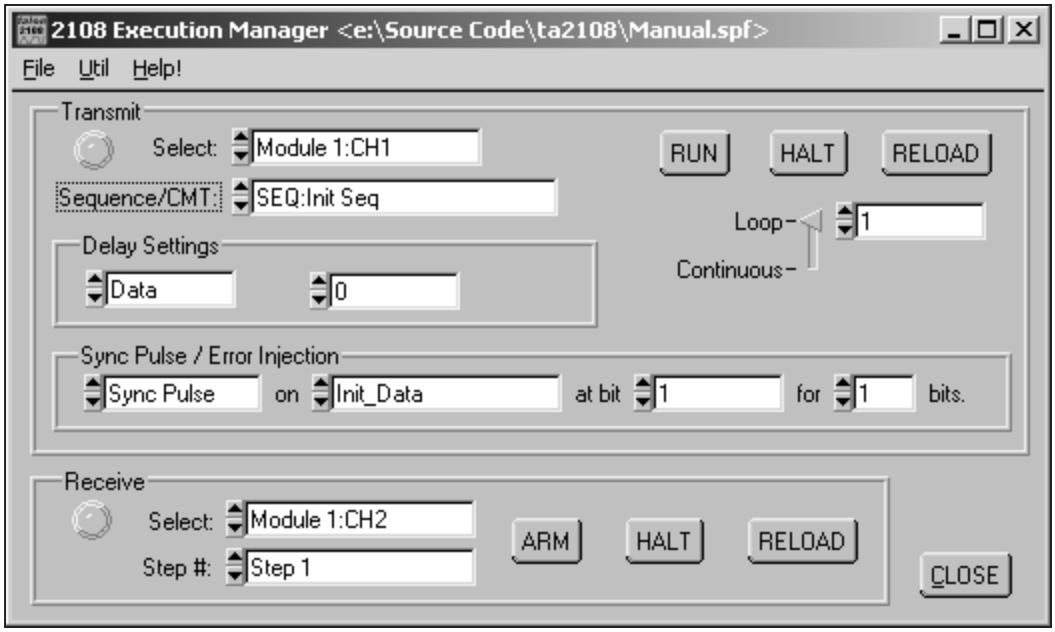
The Soft Front Panel Execution Manager is part of the VXI plug&play instrument driver and allows the user to load and execute project editor files. The Soft Front Panel is started by clicking the 2108 Front Panel program icon from the VXIpnp folder.



To open the Soft Front Panel Execution Manager panel, select "**Util**" on the menu bar.



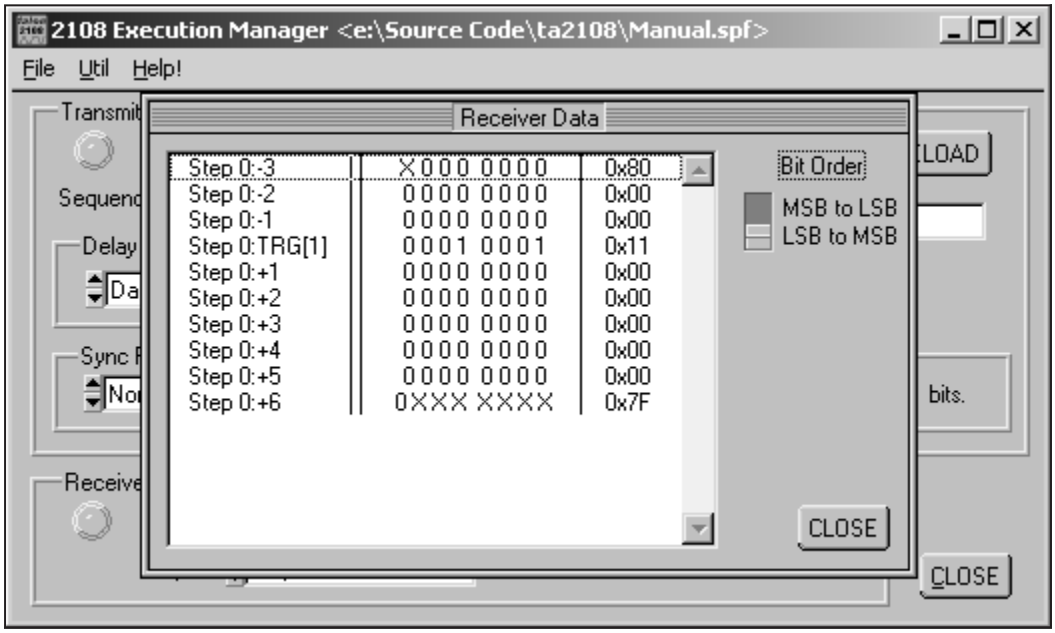
The following figure illustrates the Soft Front Panel Execution Manager panel.



The Soft Front Panel Execution Manager adds a lower level view of the 2108 hardware while still providing all the features of the Development System Execution Manager.

The differences between the two execution managers are listed below.

1. SFP Execution Manager has a “Delay Settings” section that allows the user to change the transmitter signal delay interactively.
2. Receiver data is not automatically saved. The results of a record sequence can be saved by selecting the “**File**” menu. Record data up to 1024 records can be viewed by selecting the “**Util**” menu.



3. Separate “**Halt**” and “**Reload**” buttons.
4. Transmit LED Description:

   Off Transmitter is IDLE or STANDBY.

   Yellow Sequence is running.

Green Transmitter is outputting a CMT.

5. Receiver LED Description:

Off Receiver not recording.

Yellow Receiver is ARMed and waiting for a trigger.

Green Receiver is recording post trigger data.

4.5 Evaluate and Analyze Results

After executing a transmit channel or arming a receive channel, it is often necessary to determine the status of the instrument to determine if the operation was successful or not.

The following functions return the channel status:

ta2108_GetStatus

ta2108_txQueryStatus

ta2108_rxQueryStatus

4.5.1 Transmitter Status

The following table illustrates the transmitter status word.

Bit #															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU		CERR	MB	MA	RTN	SUB	BSGB	FULLB	GRNTB	MSEL	BSGA	FULLA	GRNTA	RUN	BUSY

Table 4-1 Transmitter Status Bits

Bit Description:

BUSY This bit is set high when the transmitter is outputting a user defined CMT.

RUN This bit is set high when a test sequence/subroutine is running.

GRNTA This bit is set high when Bank A memory is available to the user for programming.

FULLA This bit is used for Bank A/BankB continuous output applications.

BSGA This bit is set high when Bank A memory is available to the transmitter for outputting.

MSEL This bit is set low when the transmitter is outputting data from Bank A and high for Bank B.

GRNTB This bit is set high when Bank B memory is available to the user for programming.

FULLB This bit is used for Bank A/BankB continuous output applications.

BSGB This bit is set high when Bank B memory is available to the transmitter for outputting.

SUB This bit is set high when the subroutine stack exceeds sixteen.

RTN This bit is set high when a RETURN command is executed without a matching subroutine.

MA This bit is set high when the transmitter is instructed to output data from Bank A memory without BSGA set high.

MB This bit is set high when the transmitter is instructed to output data from Bank B memory without BSGB set high.

CERR This bit is set high when the transmitter sequence clock is less than 122Hz

4.5.2 Receiver Status

The following table illustrates the receiver status word.

Bit #															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU		STEP				MERR	CERR	HFCRL	LFCRE	LFCRS	LFCRF	LSTEP	WAIT	ARM	BUSY

Table 4-3 Receiver Status Bits

Bit Description:

BUSY This bit is set high when the receiver is recording post trigger data.
ARM This bit is set high when the receiver is recording pre trigger data.
WAIT This bit is set high when the record sequence is waiting for the TxAck signal from the adjacent transmitter.
LSTEP This bit is set high when the last step of a record sequence is active.
LFCRF This bit is set high when the specified bit rate for the Low Frequency Clock Recovery logic is too fast.
LFCRS This bit is set high when the specified bit rate for the Low Frequency Clock Recovery logic is too slow.
LFCRE This bit is set high when the Low Frequency Clock Recovery logic detects an error.
HFCRL This bit is set high when the High Frequency Clock Recovery logic is locked.
CERR This bit is set high when the transmitter sequence clock is less than 488Hz
MERR This bit is set high when the receiver tries to record data to memory that is not released.
STEP These bits return the active record sequence step number.

4.5.3 Receiver Data

The following functions are used to download data from the receiver memory:

ta2108_GetReceiveDataArray

ta2108_GetReceiveDataFile

ta2108_rxQueryRecordDataArray

ta2108_rxQueryRecordDataFile

Once the data has been downloaded from the receiver memory the following functions can be used to query the results:

ta2108_rxQueryNumOfEvents

ta2108_rxQueryEvent

ta2108_rxQueryEventData

4.5.3.1 Receiver Data Format

The data format is comprised of multiple 32 bit records. The data encoding of these records are described below:

Bit #																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE DATA												TYPE				GOOD1								GOOD0							

Table 4-2 Record Data Format Bit Description

Field Description:

GOOD0 Eight bits of good zero record data (1 = valid low level). The data is stored MSB to LSB, i.e., the MSB position contains the oldest data bit and the LSB contains the newest data bit.

GOOD1 Eight bits of good one record data (1 = valid high level). The data is stored MSB to LSB, i.e., the MSB position contains the oldest data bit and the LSB contains the newest data bit.

TYPE Four bits that define the record type. There are five record types that return trigger information used to reconstruct the serial data stream.

TRIGGER (0x3)	The trigger record contains the trigger number and trigger size.
TIME STAMP (0x5)	Each time stamp record contains four bits of the 32 bit time stamp. The time stamp value represents the number of receiver clocks that have elapsed from the beginning of the record sequence.
POST (0x6)	Indicates post trigger data. No additional data.
SYNC (0x7)	Indicates that the last bit which triggered this sequence step is located in the previous record. Bits 23 to 16 of the prior record indicates the LSB position in the GOOD0/GOOD1 data that caused the trigger to be true (TBIT). The sync record also contains the sequence step number.
EOR (0xF)	Indicates the end of record. No additional data.

TYPE DATA Twelve bits of data used for data reconstruction.

TRIGGER	Bit 31 to 24 indicates the trigger size. Bits 23 to 20 indicates the trigger number.
TIME STAMP	Bits 23 to 20 indicates the next four bits of time stamp data. Eight time stamp records make up the 32 bit time stamp value (LSN to MSN).
SYNC	Bits 23 to 20 indicates the record sequence number.

Each record contains the next eight samples of serial data. The GOOD0/GOOD1 data indicates the sample value recorded, bit zero is paired with bit eight, bit one with bit nine, etc. The following table shows the sample value based on the GOOD0/GOOD1 value.

GOOD1	GOOD0	Sample Value
0	0	Sample bit between VIL and VIH references
0	1	Sample bit less than VIL reference
1	0	Sample bit higher than VIH reference
1	1	Invalid data

GOOD0/GOOD1 data prior to the SYNC record up to the TBIT is pre-trigger. GOOD0/GOOD1 data after the TBIT including SYNC, TRIGGER, TS and POST records are post-trigger.

4.5.3.1.1 Record Data Format Example

The example record data file was generated by setting up the 2108 transmitter to output the following CMT:

Table size: 16 Bit Order: MSB Parity Control: Reset Parity: Default Bank: A

	M1	M2	Type	Waveform	#Bits	Data	AP	Description
1	L	L	Data		8	0x00	☐	
2	L	L	Data		8	0x11	☐	
3	L	L	Data		8	0x22	☐	
4	L	L	Data		8	0x33	☐	
5	L	L	Data		8	0x44	☐	
6	L	L	Data		8	0x55	☐	
7	L	L	Data		8	0x66	☐	
8	L	L	Data		8	0x77	☐	
9	L	L	Data		8	0x88	☐	
10	L	L	Data		8	0x99	☐	
11	L	L	Data		8	0xAA	☐	
12	L	L	Data		8	0xBB	☐	
13	L	L	Data		8	0xCC	☐	
14	L	L	Data		8	0xDD	☐	
15	L	L	Data		8	0xEE	☐	
16	L	L	Data		8	0xFF	☐	

The 2108 receiver record sequence was programmed below:

Triggers: 1 @ 64 bits Sequence Steps: 1

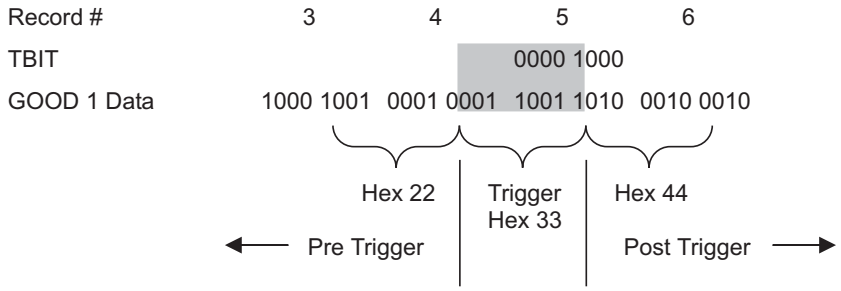
	Triggers					tecord Se
	Q1	Q2	Size	Pattern	Value	Step 1
Trigger 1	X	X	8	0011 0011	0x33	☑
PreTrigger						24
PostTrigger						96
Wait TxAck						☐
Loop						1
Last Step						☑

The following table shows the resulting data recorded.

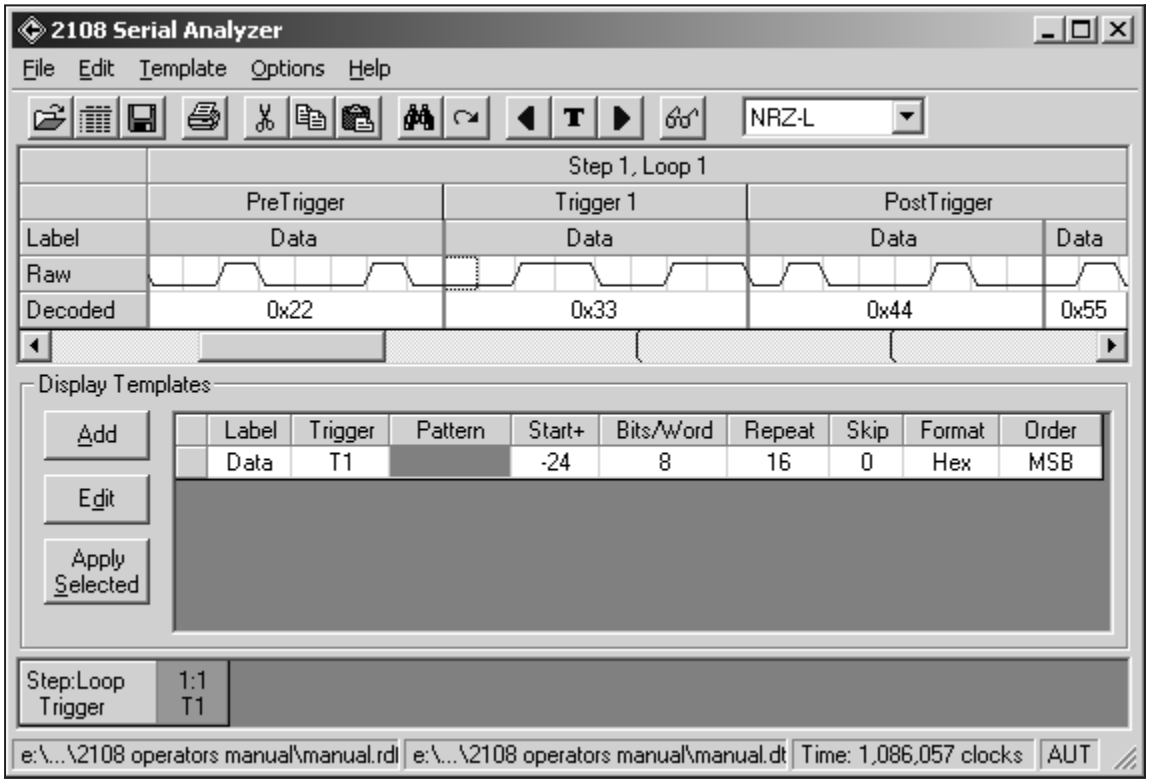
Record #	Data (Hex)	Decoded Data (Binary)				
		Type Data	TYPE	GOOD1	GOOD0	Notes
1	00 00 00 FF	-	-	0000 0000	1111 1111	
2	00 00 00 FF	-	-	0000 0000	1111 1111	
3	00 00 89 76	-	-	1000 1001	0111 0110	
4	00 00 11 EE	-	-	0001 0001	1110 1110	
5	00 08 9A 65	0000 1000 (TBIT)		1001 1010	0110 0101	Bit 3 of this record caused the trigger to be true.
6	00 07 22 DD	0000 (Sequence step 1)	0111 (SYNC)	0010 0010	1101 1101	Previous record contained the trigger bit, add 1 for step.
7	08 03 AB 54	0000 1000 (Trigger size 8), 0000 (Trigger 1)	0011 (TRIG)	1010 1000	0101 0100	Add 1 for trigger.
8	00 95 33 CC	1001 (TS bit 3-0)	0101 (TS)	0011 0011	1100 1100	TS = 9h
9	00 65 BC 43	0110 (TS bit 7-4)	0101 (TS)	1011 1100	0100 0011	TS = 69h
10	00 25 44 BB	0010 (TS bit 11-8)	0101 (TS)	0100 0100	1011 1011	TS = 269h
11	00 95 CD 32	1001 (TS bit 15-12)	0101 (TS)	1100 1101	0011 0010	TS = 9269h
12	00 05 55 AA	0000 (TS bit 19-16)	0101 (TS)	0101 0101	1010 1010	TS = 0 9269h
13	00 15 DE 21	0001 (TS bit 23-20)	0101 (TS)	1101 1110	0010 0001	TS = 10 9269h
14	00 05 66 99	0000 (TS bit 27-24)	0101 (TS)	0110 0110	1001 1001	TS = 010 9269h
15	00 05 EF 01	0000 (TS bit 31-28)	0101 (TS)	1110 1111	0000 0001	TS = 0010 9269h

16	00 06 77 88	-	0110 (POST)	0111 0111	1000 1000	
17	00 06 F8 07	-	0110 (POST)	1111 1000	0000 0111	
18	00 06 00 FF	-	0110 (POST)	0000 0000	1111 1111	
19	00 0F FF FF	-	1111 (EOR)	1111 1111	1111 1111	

The TBIT in record 5 locates the last bit of the trigger as bit 3. Since the trigger size is eight from record 7 we know that the trigger data starts in bit 2 record 4 as illustrated below.



The following figure illustrates the 2108 Serial Analyzer display for this example.



4.6 Close the VXI session.

All 2108 session handles must be closed at the end of the users application.

If the “ta2108_LoadProject” function was used to initialize the 2108 then the “***ta2108_CloseProject***” will close all the session handles.

If any of the other functions were used to initialize the 2108 channels, refer to section 4.1, then the "***ta2108_close***" function must be used for each channel.

WARNING
Failing to close a channel session will result in allocated memory not being released.

Appendix A Glossary

Bipolar	One signal represents a state.
Comparator	Compares an input signal with a reference level
Differential	A pair of signals represent a state. When one is at a high level the other is at a low level and vice-versa.
Driver fault	The output driver (for S1 thru S8) is shut down due to current overloading or over temperature
ECL	Emitter coupled logic
TxFlagIn	An input signal which the transmitter can query
TxFlagOut	An output signal which the transmitter can set
Good “0”	A signal generated when an input signal is less than VIL
Good “1”	A signal generated when an input signal is greater than VIH
HFCR	High Frequency Clock Recovery
LED	Light Emitting Diode
LFCR	Low Frequency Clock Recovery
Marker	An output signal used to mark one or more portions of the output data stream
Reference	A programmable DC voltage
RX01	Model 1 Receiver Interconnect Module
RxArm	Record sequence “ARMed” and waiting for a trigger
RxBusy	Record sequence running
RxClkOut	An output signal of “Clock In” used to align data with clock
RxG0Val	An output signal of “Good-0” used to align data with clock
RxG1Val	An output signal of “Good-1” used to align data with clock
RxTrigNum	The particular trigger number which occurred for a particular trigger event
RxTrigValid	Trigger valid signal generated by the receiver when a trigger has occurred
RxWait	Receiver is waiting for an acknowledge of a Trig Valid
Slew rate	Rate of change of an output transition (typ in V/ns)
SMA	A small screw-on RF connector
TxSyncPulse	An output pulse which can be positioned in relation to the output data stream (typically used to synchronize another instrument).
Trinary	Three distinct levels
Tri-state	A third state indirectly occurring when not being driven to a defined state.
TX01	Model 1 Transmitter Interconnect Module
TxAck	A signal from the transmitter sent to the receiver acknowledging that a trigger valid test true has occurred
TxBusy	Transmitter running
TxClkOut	A time adjustable (also invertible) representation of the clock used by the transmitter
UUT	Unit Under Test
V+	Positive supply voltage provided by the user
V-	Negative supply voltage provided by the user
VOH	A reference defining an Output High Level
VOL	A reference defining an Output Low Level

VOZ	A reference defining an Output third state level.
VIH	A reference defining an Input High Level
VIL	A reference defining an Input Low Level
VXI	VME Extensions for Instrumentation

Appendix B Serial Bus Interface Example

Although there are many synchronous serial interface structures in use, all contain a **DATA** signal. The data signal transmits information to the destination. The information could contain elements such as headers and error-checking information (i.e., parity or checksum).

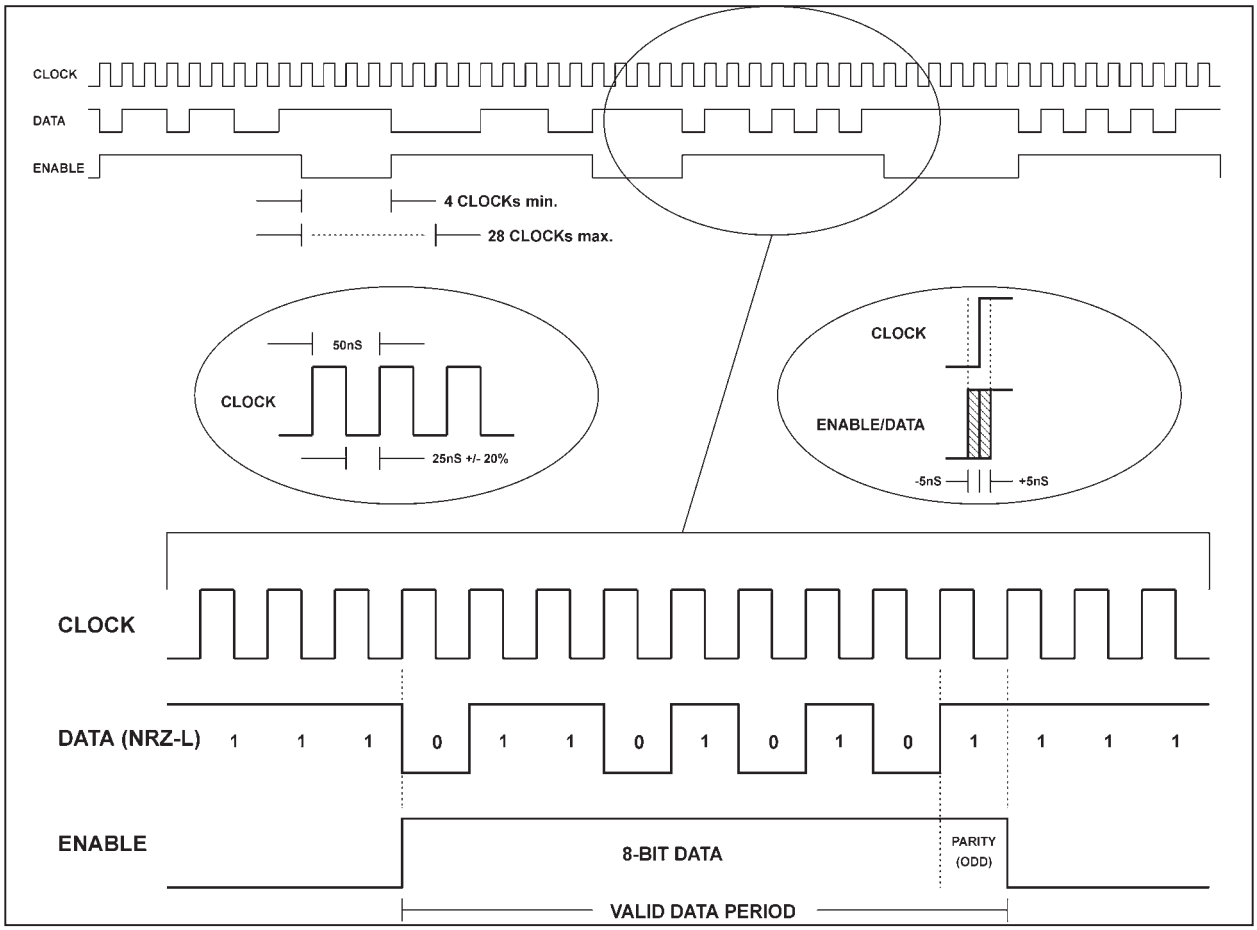
In addition to a **DATA** signal, some serial interfaces are comprised of other signals such as **CLOCK** and **ENABLE** which are used to qualify the **DATA** signal.

The **CLOCK** signal produces a continuous clock waveform and serves to synchronize the occurrence of each **DATA** bit with a specific and constant time period. Typically each bit of **DATA** corresponds directly to one complete **CLOCK** cycle.

The **ENABLE** signal identifies the occurrence of a DATA transmission and can track either each specific **DATA** word (i.e., DATA ENABLE) or a group of **DATA** words (i.e., FRAME ENABLE). **DATA** is considered valid only while the **ENABLE** signal is "true".

The first step in emulating any interface is to understand what elements are involved and what resources can be assigned to them. This step also includes understanding the relationships between these elements as they perform in a real-world situation.

For the particular serial bus in this example, it is known that there will need to be output channels for the CLOCK, DATA and ENABLE signals. The bus uses 422/485 type drivers. Looking at the timing diagrams below, the following relationships can be surmised:



a) The CLOCK frequency is 20MHz with a 50% duty cycle (25nS "low" period +/-20%).

b) ENABLE goes "high" to indicate the valid DATA period.

c) ENABLE and DATA transition upon the "rising edge" of CLOCK within +/-5nS.

d) Each DATA bit equates to one CLOCK period.

e) The valid DATA word length is 8 bits + 1 odd parity bit (i.e., 9 CLOCK periods total).

f) CLOCK is continuous and free-running.

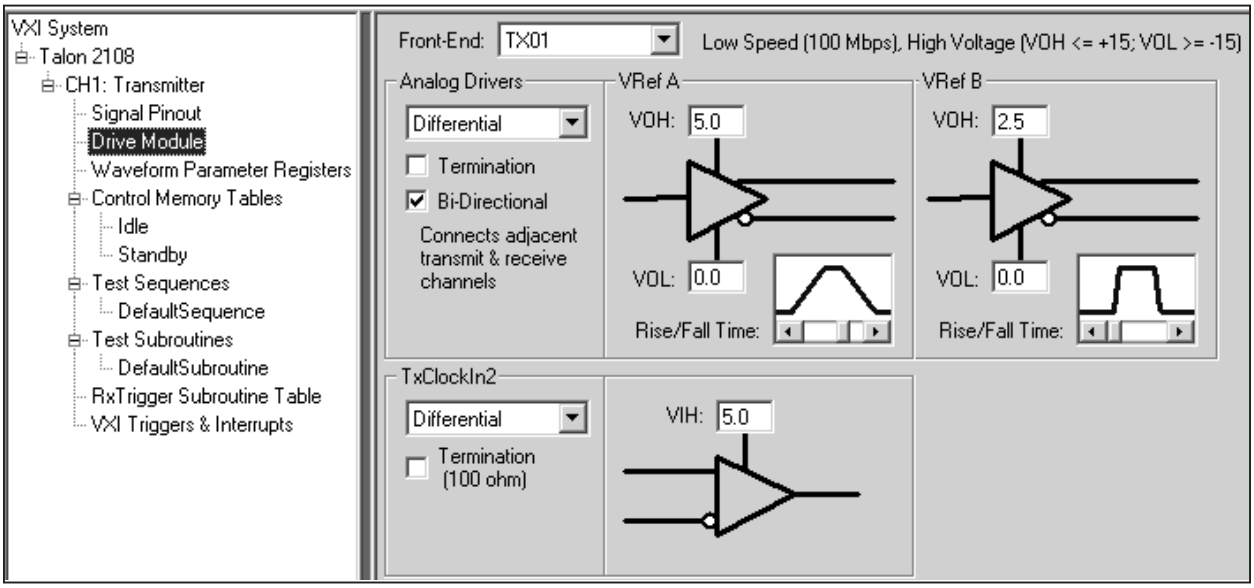
g) DATA is held at a logical “high” during ENABLE “low”.

h) “Invalid” DATA periods are variable (i.e., when ENABLE is “low”) from a minimum of 4 CLOCK periods to a maximum of 28 clock periods.

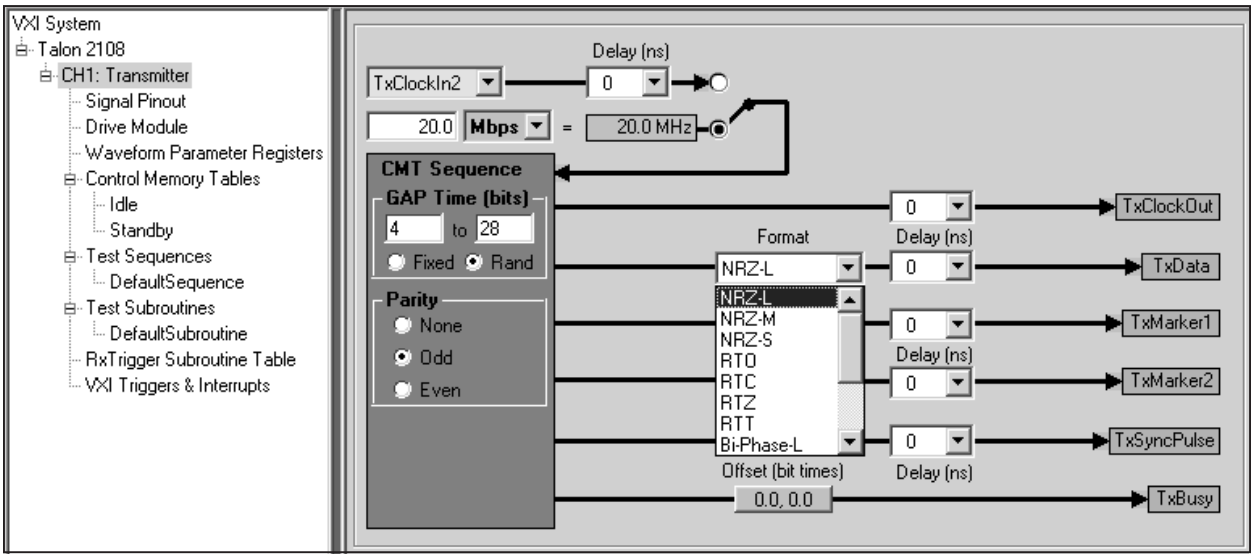
i) Data Format is NRZ-L.

Using the Model 2108TX Project Development Software, a user can easily implement a variety of serial bus emulation scenarios. Configuring setup parameters and constructing data streams are accomplished via graphical panels.

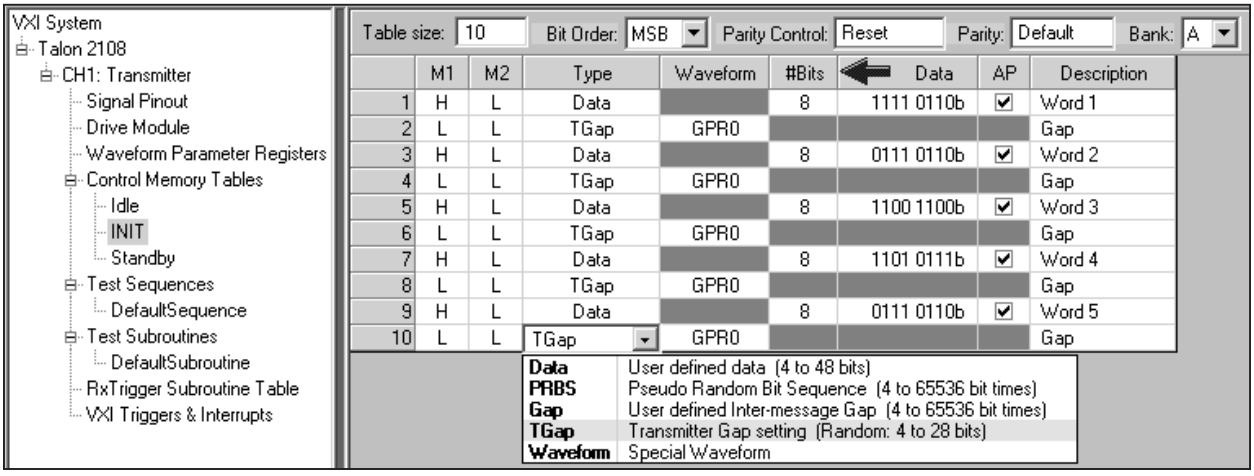
The first step in setting up the example serial interface is to program the electrical characteristics. Since this is a 422/485 serial interface we know the driver characteristics include differential signals, 0-5Vdc for the logic levels and with a low to mid-level slew rate. The Drive Module panel is used to program the electrical characteristics for both Ref. A (primary) and Ref. B (error).



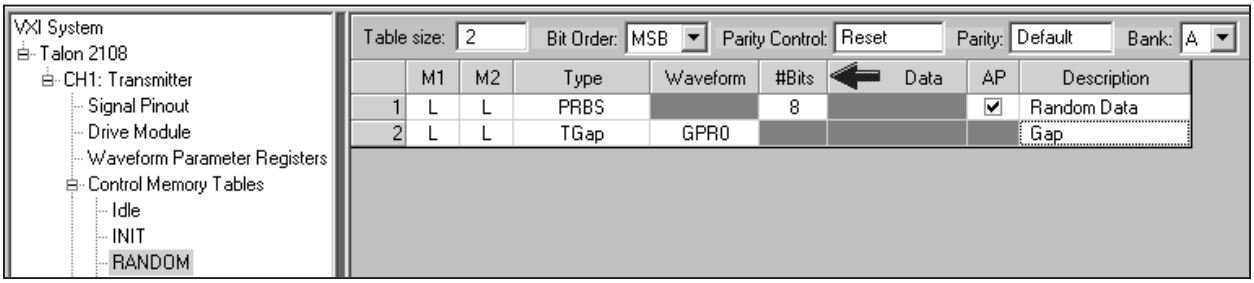
The logical characteristics of the example are programmed by use of the Transmitter panel. Using the information from our timing diagrams and the fact that it is a 422/485 interface we know the data rate, data gap min/max values, logical format and parity type.



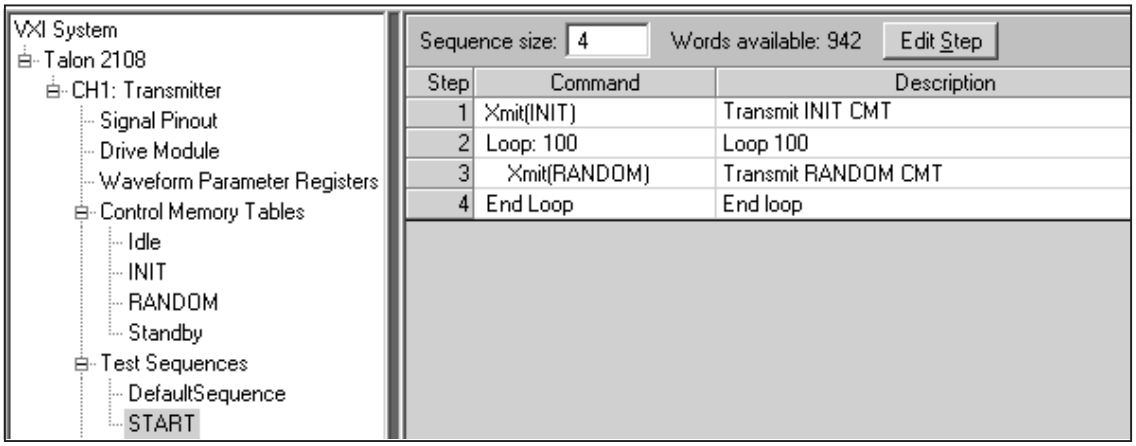
The next step after programming the electrical and logical characteristics is to prepare data files to be transmitted to the UUT. Data files are stored as Command Memory Tables, (CMT's). Each command memory table is programmed as a file containing the data and the parameters pertinent to that data file. Since our example is quite simple our first CMT will be named INIT and contain 5 8-bit words. The bit order is set to MSB and the TGap generator is programmed to use gap waveform GPRO. Parity is appended after each word. Marker 1 (M1) is programmed to be output High during each data word transmission and Low during gap periods to match the Enable Signal operation.



The second data table to be used in our example will be programmed to output Pseudo Random Bit Sequence data instead of fixed data. The Control Memory Tables panel will be used to create a table named RANDOM.



The next step in programming our example serial bus is to define the output sequence of the data files we have created. Output sequences are defined in the panel named Test Sequences. In our example, we will define a simple sequence to output the CMT named INIT followed by a loop sequence outputting the CMT RANDOM looped 100 times. The Test Sequence name is START.



The final step in the creation of a serial bus emulation test is to connect to the "real" thing. After performing the physical interconnect the CMT's and/or the Test Sequences are downloaded to the Model

2108TX from the VXI controller. This may be accomplished in one of two ways. In the first method, the saved file created by the Project Development Editor may be interactively downloaded using the 2108TX Execution Manager. The Execution Manager is a part of the Project Development Editor. The Execution Manager provides interactive panels to download, start and run individual CMT's or Test Sequences. Control is provided for single pass or continuous loops to aid in debugging problems. The second method is to use the 2108TX VXIplug&play drivers to download and control the execution of the CMT's or the Test Sequences. The Execution Manager and the VXIplug&play drivers are WIN95, 98 and NT compatible.