# REQUIREMENTS AND FUNCTIONAL SPECIFICATION

## EVLA Correlator Chip Test Analysis Software

**A software tool that decodes and analyzes correlator chip output data frames, produces spectra, and compares functional output with behavioral output**

RFS Document: **A25082N0011**

Revision: 1.2

Frances Lau, July 16, 2003

*National Research Council Canada*
*Herzberg Institute of Astrophysics*
*Dominion Radio Astrophysical Observatory*

*P.O. Box 248, 717 White Lake Rd*
*Penticton, B.C., Canada*
*V2A 6K3*

# Table of Contents

# List of Figures

# List of Tables

# 1 Revision History

| Revision | Date | Changes/Notes | Author |
|---|---|---|---|
| 1.0 | June 11, 03 | Initial Revision | Frances Lau |
| 1.1 | June 19, 03 | Revision | Frances Lau |
| 1.2 | July 16, 03 | Revision to give additional details regarding record keeping methods. | Frances Lau |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# 2 Introduction

This document describes the requirements and design concepts for a piece of software that will act as a tool in the testing of the correlator chip. By decoding and analyzing the correlator chip output data frames, the software will produce spectra and display a diagram of the correlator chip highlighting the activated data paths. Functional output will be compared with behavioral output generated by a simulator written in C. It will be possible to print and save the graphs and the chip diagrams for record keeping purposes.

The software will have a Windows-based graphical user interface. It will be developed using Microsoft Visual Studio, primarily Visual Basic .NET.

# 3 Context

This software will act as a tool in the testing of the correlator chip. The spectra produced by the software will help the person performing the tests quickly verify that the chip is meeting the functional requirements for that test case. The diagram of the activated data paths in the chip will help the user visualize which areas of the chip were tested and enable him/her to develop effective test cases.

# 4 Overview

Figure 4-1 is a block diagram that presents an overview of this Correlator Chip Test Analysis Software.

The present testing module `corr_chip_top_tester.v` uses the test cases defined in the test bench file `corrchip_testcases.v` to test the correlator chip. It outputs two files: `mcb_reg*.txt` and `lag_frames*.txt`.

The `mcb_reg*.txt` file is an ASCII representation of the contents of the MCB (Monitor & Control Bus) registers. The `lag_frames*.txt` file is an ASCII representation of the data transmitted to the LTA (Long Term Accumulator) Controller. Data is transmitted to the LTA Controller in data frames, where each frame is the control, status, and data for one CCC (Correlator Chip Cell). An independent frame is transmitted for each CCC [1].

This Correlator Chip Testing Analysis Software will decode the `mcb_reg*.txt` and `lag_frames*.txt` files to produce a diagram of the chip showing the activated data paths, as well as complex lag data plots, Fourier transform plots, and difference plots that indicate the difference between the correlator chip functional simulation output and the behavioral simulation output.

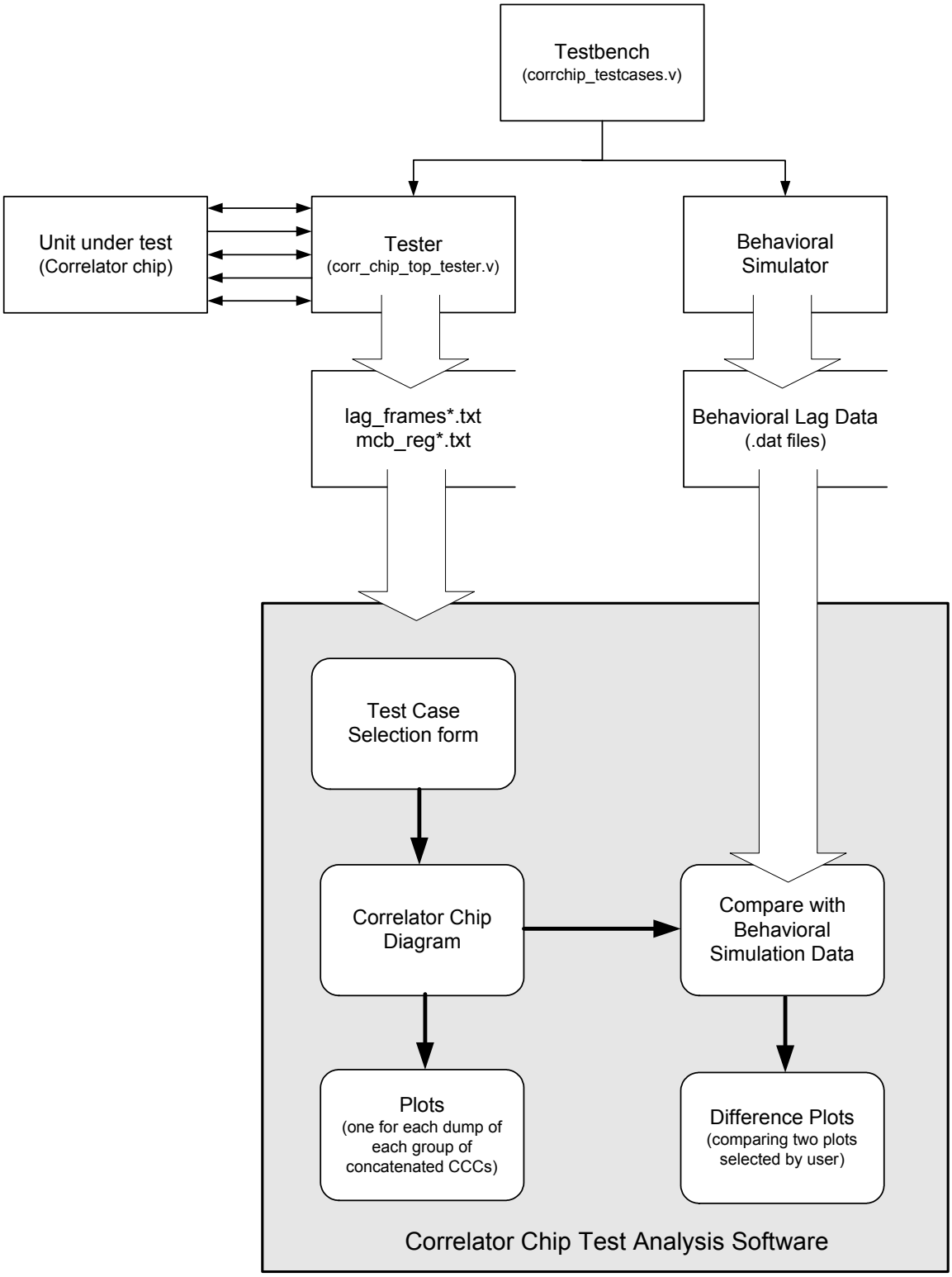


Figure 4-1: Overview of Correlator Chip Test Analysis Software

# 5 Requirements

The following is a list of the requirements for this test analysis software.

## 5.1 Functional Requirements

1. As shown in Figure 5-1, the software will display a simplified diagram of the correlator chip that shows all the input switches. It will highlight on this diagram which data paths are being used.

   Active concatenated CCCs will be grouped together using a yellow border. The user will be able to click on one of these groups to view the plots for this group.

   Each group of concatenated CCCs may generate multiple dumps. The user will be able to view the plot for each dump for that group of concatenated CCCs.

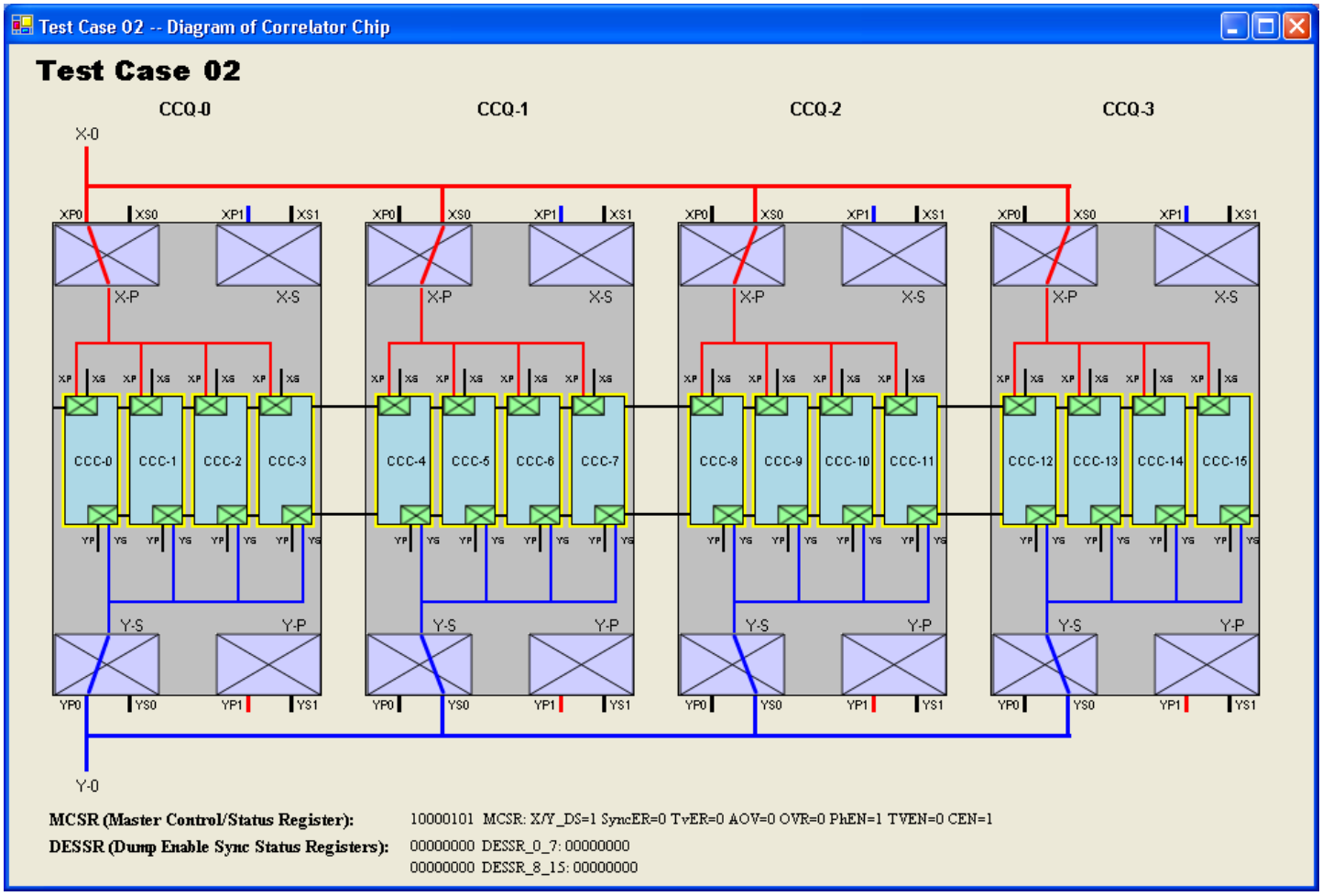


Figure 5-1: Sample Diagram of Correlator Chip window

Information for this diagram will be obtained from the mcb_reg file. Table 5-1 outlines what information from the mcb_reg file will be displayed and what format it will be displayed in.

Table 5-1: Display format of registers

| **Name of Register** | **Display Format** |
|---|---|
| | |
| MCSR | In text format. |
| CCCSCR | Graphically. The data paths will be highlighted in red/blue on the diagram. Wires will appear on the diagram to connect the inputs only when the inputs are used. |
| CCQR | Graphically. The data paths will be highlighted in red/blue on the diagram. Wires will appear on the diagram to connect the inputs only when the inputs are used. |
| XYSSR | Not displayed, but can be viewed in the mcb_reg file. |
| XSTATUS, YSTATUS | Not displayed, but can be viewed in the mcb_reg file. |
| DESSR | In text format. |

2. The following graphs will be produced:

**I) Lag plots -** The following will be generated for *each* dump of *each* group of concatenated CCCs

a) real lag plot
   (i) from correlator chip functional simulation
   (ii) from behavioral simulation
b) imaginary lag plot
   (i) from correlator chip functional simulation
   (ii) from behavioral simulation
c) magnitude lag plot
   (i) from correlator chip functional simulation
   (ii) from behavioral simulation
d) Fourier Transform – amplitude plot
   (i) from correlator chip functional simulation
   (ii) from behavioral simulation
e) Fourier Transform – phase plot
   (i) from correlator chip functional simulation
   (ii) from behavioral simulation

**II) Difference plots**

Difference plots help the user compare the functional simulation output with the behavioral simulation output generated by the simulator written in C.

The user will be able to select any two graphs generated within one test case and attempt to display the difference between them. If the graphs do not have the same domain and range, the program will output a message indicating that a difference plot cannot be produced. The graphs compared do not necessarily have to come from the same group of concatenated CCCs, but they do have to be from the same test case.

The user will be able to view the full lag_frames file and the full mcb_reg file to see the header information.

## 5.2 Performance Requirements

1. The program will operate on a PC with a Microsoft Windows operating system.

## 5.3 Environmental Requirements

1. The program will be run in a Windows environment.

## 5.4 Interface Requirements

1. The program will have a Windows-based graphical user interface.

2. The files output by the correlator chip tester program (`corr_chip_top_tester.v`) for each test case must all be in a single directory. The user will have the opportunity to specify the path to this directory.

3. The user will be presented with a list of the test cases available. Figure 5-1 is a picture of the window the user is presented with.

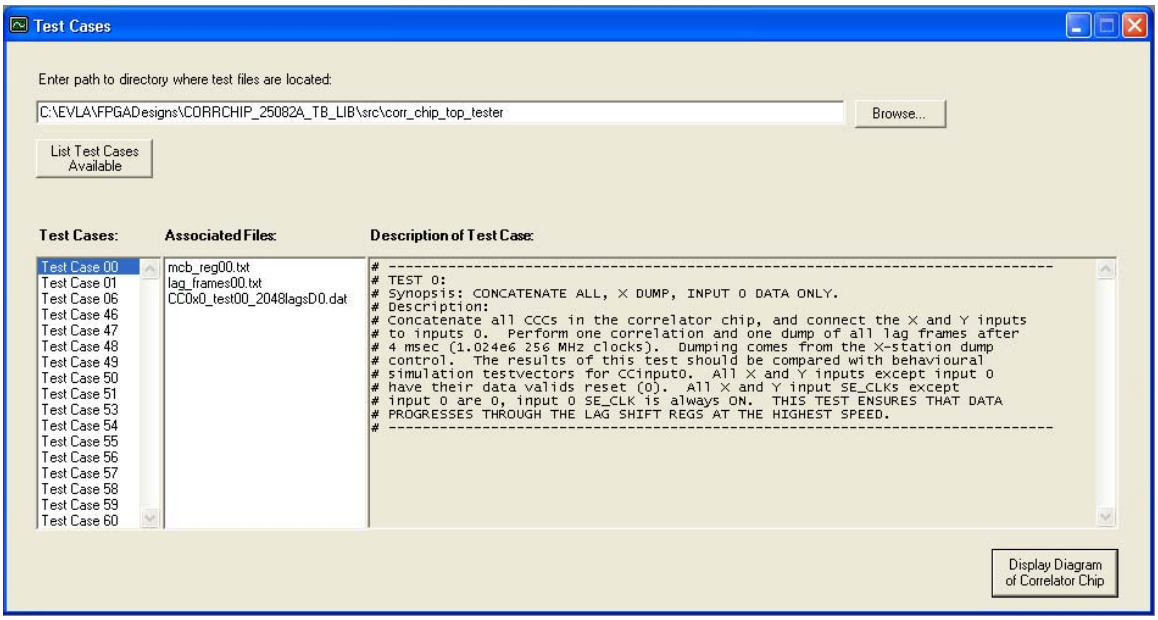


Figure 5-2: Test Cases window

4. Each test case must have one `mcb_reg` file. The `lag_frames` file is optional. The files must have the filename format `mcb_reg*.txt` and `lag_frames*.txt`.

5. After selecting a test case, the user will click the button "Display Diagram of Correlator Chip" and a new form will appear.

6. As shown in Figure 5-2, the Diagram of Correlator Chip form will display a simplified diagram of the correlator chip that emphasizes the input switches in the chip. It will highlight the flow of data through the chip by displaying wires connecting only the inputs that are used.

7. Each group of concatenated active CCCs will be highlighted separately in yellow. The user will click one group of concatenated CCCs to view the plots for this group. He/she will be directed to a new window with a tree diagram listing all the possible plots.

# 6 Functional Specifications

The organization of this software package will be based on the types of Windows forms presented to the user. The main forms are divided into three categories based on function.

1) Test case selection form
2) Correlator chip diagrams
3) Plots

All forms will be designed using Windows Forms, the new platform for Microsoft Windows application development, in the Microsoft Visual Studio .NET Framework.

### 1) Test case selection form

Three main tasks will be executed through this form. First the program will take the directory path specified by the user and list all the possible test cases located in this directory. This will be implemented using the Visual Basic `Dir$()` function. Possible test cases are indicated by the existence of a `mcb_reg*.txt` file in the directory. A `lag_frames*.txt` file is not mandatory.

Secondly, when the user selects a test case, the program will open the associated `mcb_reg*.txt` file and display the description of the test case. File I/O will be implemented using the `StreamReader` objects in the `System.IO` namespace. This requires an `Imports System.IO` statement at the top of the class. Using `StreamReader` objects is the recommended approach for file I/O using the .NET framework.

Finally, the program will check the directory again to see if an associated `lag_frames*.txt` exists, and then list all the files associated with this test case. This will also use the Visual Basic `Dir$()` function.

### 2) Correlator chip diagrams

One correlator chip diagram will be displayed for each test case. The program will decode the `mcb_reg` file for this particular test case to determine which switches are on, which inputs are used, and which CCCs are concatenated.

**Eight graphics files will be needed by this program. They will be automatically installed in a folder called `graphics` located in the same directory as the Correlator Chip Test Analysis Software executable file.** If the required graphics files are not found, a detailed error message will be displayed.

For now, the MCSR will simply be displayed in text format. Later versions of this software may be capable of decoding the MCSR and displaying the bits graphically.

The CCCSCR and CCQR registers will be used to determine which switches are on and which CCCs are concatenated.

Since the `printForm` method is not available in Visual Basic .NET, printing and saving will be implemented by using the ALT-PrintScreen keystroke to copy an image of the current active window to the clipboard. Since this application will be a Multiple Document Interface (MDI) Application, all the open windows in the application are considered the current active windows, and this causes a problem since we only want the Chip Diagram window. To solve this problem, the program will temporarily change the Chip Diagram form into a non-MDI child form, execute the ALT-PrintScreen keystroke, and then change the Chip Diagram form back into a MDI child form.

**3) Plots**

All plots will be displayed using the MSChart control. **Therefore, the files `AxInterop.MSChart20Lib.dll` and `Interop.MSChart20Lib.dll` must be distributed with the application.**

One set of plots will be available for each dump of each group of concatenated CCCs. Further information specific to a particular type of plot is given below.

Functional simulation plots:

The program will open the `lag_frames` file associated with this test case and decode it to determine if the file contains multiple dumps for this group of concatenated CCCs. The dumps will be differentiated by timestamp. If there are multiple dumps, the user will be able to select which dump he/she wants a graph displayed for.

The program will extract the data frames that correspond to the selected plot and decode the frame to obtain the required data.

The data will be processed to obtain the desired plot.

The data will be stored in an array. This array will be passed to the MSChart control, which will be used to display the graph.

Behavioral simulation plots:

Behavioral simulation data is stored in numerical format in `.dat` files, with the real component in one column and the imaginary component in another column. The data will be extracted from the file, stored in an array, processed, and passed to the MSChart control.

Fourier transform plots:

The complex lag data will be stored in a text file. This text file will be passed to a FFT program written in C/C++. **A copy of the FFT program will be automatically installed in a folder called `fft`, located in the same directory as the Correlator Chip Test Analysis Software executable file.** If the required FFT executable file is not found, a detailed error message will be displayed.

The FFT program is executed with the command:
`fftPath window sourceFilePath numLags`,
where `fftPath` is the path to the fft.exe file
`window` is an integer specifying whether or not windowing should be turned on or off (1 = on, 0 = off).
The window function is: $Wn(n, N) := 0.5 [ 1 - \cos(2\pi n / N)] + 1 \cdot 10^{-18}$,
where n = lag number
N = number of lags
`sourceFilePath` is the path to the file where the complex lag data is stored, and
`numLags` is the number of complex data points to convert.

Inside the FFT program, the bandwidth of the output data is set to 1 MHz. This value is used to normalize the data bins to power/MHz.

The output of the FFT program will be sent to a text file and this Correlator Chip Test Analysis Software will extract the data from the text file, process it, store it in an array, and display it using the MSChart control.

The program will create temporary files to store the data. The temporary files will be saved in the temp folder, which is located in the same directory as the Correlator Chip Test Analysis Software executable file. Therefore, the user must have permission to write to the location where the software is installed.

Difference plots:

A record of all plots created for a particular test case will be kept by a `FrmPlotList` form. The user will be able to select any two plots from this record and attempt to create a difference plot using them.

If the two plots are not of the same type or if they have a different domain or range, an error message will be displayed. The domain comparison is based on the original graph, before any zooming has occurred.

The program will be able to calculate the mean difference and the standard deviation around the mean for the plot. The mean difference is calculated as the sum of all the value of the data points in the difference plot divided by the number of data points.

Zooming:

To zoom in or out, the user will select one of the zooming icons at the top of the graph, and then double-click on a data point. The program will zoom in or out horizontally by a factor of 2 around this data point. For zooming in, this will be implemented by copying the data points within (50%/2) of the displayed domain around the selected data point. For zooming out, this will be implemented by copying the data points within (200%/2) of the displayed domain around the selected data point.

If Manually Adjust Y-Axis Scale is not checked, the y-axis scale will automatically adjust itself to fit the data, while always keeping the y-axis zero marker visible. To zoom into the graph even more, the user should check Manually Adjust Y-Axis Scale. The graph will zoom into the selected data point vertically as well as horizontally by a factor of 2.

The disadvantage of having Manually Adjust Y-Axis Scale checked is that the y-axis scale will not automatically adjust itself to fit the data. Only a strict zoom by a factor of 2 will occur, so it may take longer to zoom into an area of the graph.

It is recommended that Manually Adjust Y-Axis Scale be initially left unchecked. The user should only check this option if he/she has zoomed in several times and it is apparent that it would be better if the y-axis zero marker were not displayed, that is, if the area he/she wants to zoom into is far away from the zero level.

# 7 References

[1] Carlson, Brent, Requirements and Functional Specification: EVLA Correlator Chip, July 26, 2002.