EECS 373 WINTER 2004

Lab 1: Introduction to Lab Hardware and the Logic Analyzer

Posted: January 5, 2004

Requirements

Pre-lab: There is no pre-lab for lab 1.

In lab: You must demonstrate achievement of specific milestones to your lab instructor as noted in the lab procedure below.

Post-lab: There is no post-lab for lab 1. The lab demonstration sheet is due anytime during open lab, Friday January 9, 2004.

Lab Value: 2% of total grade

Objectives

The purpose of this lab is to provide a basic introduction to the 373 lab hardware environment and introduce you to the most powerful debugging tool at your disposal, the logic analyzer. Specifically, in this lab you will:

1. Set up your software environment for the 373 lab.
2. Learn about the target hardware used in the EECS 373 lab.
3. Get reacquainted with the Xilinx Foundation Software from EECS 270
4. Learn how to generate a Xilinx bitfile from a schematic diagram and download it to the FPGA on the EECS 373 target board.
5. Learn the basic operation of a logic analyzer.
6. Practice setting triggers on the logic analyzer to capture interesting events across multiple digital signals.

Overview

Lab Software Environment

Each lab station has a Windows PC to run development and debugging tools (primarily Xilinx Foundation Series for hardware development and SDS SingleStep for software development and debugging). Signon to the 373 PCs with your CAEN uniquename and password under the ENGIN domain. A logon script will map a network drive to 373 class directory space on an Windows server maintained by CAEN with the path \caen-mahogany\DFScourse\W03\eecs373. All your project files for the class should be stored in your class directory space. You can also access your class directory space in CAEN labs. Another network drive is mapped to \caen-mahogany\DFScourse\Perm\eecs373 where files you will need for various labs are located.

You will probably want to share your class space with your lab partner. To change the permissions on your class directory so your partner can access it, see appendix A ‘Sharing NTFS Space’.

Lab Target Hardware

The target microprocessor-based hardware you will use in lab this semester consists of three, stacked, connected boards. The bottom and top boards are standard development boards sold by Motorola to companies developing products around the MPC823. The top board is the MPC823ADS daughterboard, which contains the MPC823 processor (inside the giant socket). This board is also connected to the logic analyzer via several ribbon cables. The bottom board is the MPC8xxFADS motherboard, and contains the system

memory, some I/O connections, and the debugger port. As the name implies, it can be used with different daughterboards to support other devices in the MPC8xx family. In a standard configuration, the daughterboard plugs directly into the motherboard. Four high-density connectors arranged in a square pattern pass all of the MPC823 signals between the daughterboard and the motherboard.

For the 373 lab, we have developed a custom board that sandwiches between these two boards and taps off of all the signals being transferred on these connectors. This board contains a Xilinx FPGA, switches, LED indicators, memory, and analog to digital and digital to analog converters. There is a socket for a second FPGA that is currently unused. The board is designed so that the FPGAs can be programmed to interface the MPC823 to these devices and to an external connector for debugging and for interfacing devices not on the board. Schematics of the board are available on the class web site.

The Logic Analyzer

A logic analyzer (LA) is an instrument that allows you to observe many digital signals simultaneously. The LA displays signals as a function of time much like an oscilloscope. Generally, when monitoring digital signals, only the logical state of the signal is of interest. Consequently, signals are stored and displayed on the LA as a logical 1 or 0. The LA can also display the state of a group of signals as a numerical value. This feature is very useful when observing the MPC823's address and data busses.

The LA we use in the lab, the Hewlett Packard 16600A, can monitor 192 digital signals at once. These signals are organized into 12 pods, each containing 16 signals. You will see ribbon cables and labeled connectors dangling from the back of the LA corresponding to each of these pods. Pods 1-6 are terminated into 3 connectors and connect to the MPC823ADS board. These signals represent all the address, data, and control bus signals of the MPC823ADS. Pod 8 is terminated with 16 individual connectors labeled 0-15. This Pod is used to monitor signals on the EECS 373 I/O board by connecting to pins on the TP1 connector. Pods 7 and 9-12 are not used.

The HP16600A LA is essentially an embedded HP UNIX workstation and is operated with a fairly intuitive GUI. After the system powers up, you will be presented with the main system window. In general, an LA must be configured before it can be used to monitor signals. Each signal in each pod must be configured for its digital signal type and display characteristics. Each signal or group of signals must also be assigned a name. Basic measurement configurations have been provided for most of the labs, so you will not have to configure the LA yourself at first. Configurations can be saved and loaded through the File Manager on the main system window. After you develop some familiarity with the LA, you may wish to create your own measurement configurations or modify existing configurations. Once a measurement configuration is loaded, the LA is ready to take measurements.

Lab Procedure

Part I: Creating a Test Circuit with Xilinx Foundation Software

Step 1. Starting a Project

Start the Xilinx Foundation Software (XFS) and select New Project. Enter a project name. Use your class directory space.

Specify the parameters for your project as follows:

Type:	F2.1i
Flow:	Schematic
Device family:	XC4000E

Part: XC4010EPG191

Device speed: 3

Step 2. Adding the EECS373 Library

To properly connect to the devices on the 373 lab board, the circuits that you implement in the FPGA must use the FPGA pins that are physically wired to the signals you need. In FPGA terms, you need to *constrain* a particular signal in your design to use a particular physical pin. To simplify this process, we have developed a library of I/O macros which implement the appropriate FPGA pin constraints. Each I/O device on the EECS373 I/O board is represented as a macro. Connecting a signal in your schematic to a port on one of the macros will make the corresponding connection in your final circuit implementation by forcing the Xilinx tool to connect that signal to the appropriate physical FPGA pin.

To add the EECS373 library to your project select Project Libraries under the File menu in the XFS Project Manager Window. The Project Libraries window will appear. Look in the attached libraries view for the EECS373 library. If you find it, select it and add it. If you did not find it, select Library Manager, then select Attach under the Library heading. Browse for C:\Fndtn\Xi Lib. Select the EECS373 library and click OK. If you have properly attached the eecs373 library, it will appear in Project Libraries window under Attached Libraries. Finally, select the library and add it to your project. If you are using a CAEN machine, the EECS373 library is on the class directory perm space in the Xi Lib folder. **NOTE:** If you are browsing to attach the eecs373 library in C:\Fndtn\Xi Lib and it does not appear, it is because it has already been attached. Look in the Project Libraries window and you should find the EECS373 library.

Step 3. Adding the Test Circuit

A test circuit is provided for you. To add the circuit, select Add Source Files under Project in the XFS project window. Browse for the circuit **T1** located in \caen-mahogany\DFScourse\perm\eecs373\Lab1. You should be able to see the complete test circuit if you have included the library successfully.

Take a few minutes to study the test circuit. You will see that it consists of three macros from the EECS 373 library, representing the switches (two pushbuttons, two toggle switches, and 8 DIP switches), the LED bar graph, and the “test points” on the 373 I/O board. The test points are pins that allow you to connect the logic analyzer or another external device to your circuit.

Use the hierarchy button, represented with an H, to examine the contents of each of the macros. The switches macro contains IPAD objects representing the Xilinx input pads connected to the EECS 373 I/O board switches, as well as a pull-up resistor for each switch. You can verify that the pin constraints in the macro correspond to the respective switches by viewing the “Basic I/O devices” schematic available on the EECS 373 web site. Notice the 8-position DIP switch is wired as a bus for connection convenience. The 8-position LED bar graph macro is connected similarly.

The test points macro connects to the pins that make up the connector labeled TP1 on the board, just to the right of the upper FPGA socket. You can connect to the macro via the 15 line bus, IC_OUT[31:17]. The bus is mapped one to one to each pin on the test point connector, ie IC_OUT[31] is mapped to pin 31. The test point pins, TP1, are numbered alternatively starting with pin 0 as shown:

(upper left)	1	3	5	7	...	31	(upper right)
	0	2	4	6	...	30	

The LA test connectors are labeled 0-15 and the LA channels labeled tp0-15, so the test point bus is named TP[0-14] for consistency. Note, Tp0 is now mapped to pin 31, tp1 is mapped to pin 30 etc. Each LED-signal connection is connected to a test point.

You may recall that is not possible to connect two nets with different names. To map the bus that connects the dipswitches and bar leds to the testpoints bus, buffers are used to isolate the nets. The buffers perform no electrical function. They are use to allow connection between differently named nets.

You will use one other macro from the EECS 373 library for this lab: the PROC macro. The PROC macro represents the MPC823 itself, and provides connections to the MPC823's external signals such as the address, data, and control busses. In this lab, you will use only one signal from the PROC macro: the external bus clock, labeled "gclock1".

Step 4. Implementing the Test Circuit

Implement the design by selecting the implement icon in the project window in the XFS. You should be able to find a corresponding bit file in the project directory if the implementation was successful.

Part II: Programming the EECS 373 I/O Board FPGA

To use your circuit, the bit file produced from the XFS implementation must be downloaded to the FPGA on the EECS 373 I/O Board. The download is performed via the PC's serial port. We use a special HyperTerminal Configuration to download the file. Select the desktop icon, Xilinx Download, to invoke HyperTerminal.

Before you can download the bit file, it is necessary to reset the MPC823. This is accomplished by activating the 'SingleStep on a Chip' icon on our desktop. If this is not on your desktop, go to start > program > singlestep7.5. This activates the source level MPC823 debugger that you will use in future labs. A window will appear asking for a MPC823 source file. Check 'debug without a file' and then click OK. The next window should report the processor reset successfully.

If you fail to reset the MPC823, you will observe a partial or erratic download of the bit file. It is necessary to reset the MPC823 only once after powering on the board set.

Next activate the Xilinx download icon. Under Transfer, select 'send text file'. Browse for your bit file in your project directory. You will have to set the file type setting in the browser box to all files, i.e., "`*.*`". Double click the file or select it and click open. The EECS 373 I/O board should respond with the following message in your HyperTerminal window:

```
INIT* signal went high
..................................................
DONE signal went high after 0x____ bytes
```

The left-most LED on your bar-graph display should be dimly lit, indicating a successful download. The DIP switches should now activate the LED bar-graph display, as specified by the test circuit you implemented.

Some board sets are set up to load two FPGA's. On these boards you will see two down load patterns in the HyperTerminal window. If only one FPGA is installed on this type of board, you will see a download failure message for the missing FPGA.

Part III: Observing Logic Signals with the Logic Analyzer

In this section, you will use the Logic Analyzer to observe signals in your test circuit.

Step 1. Connecting the Logic Analyzer to the EECS373 I/O Board

Connect Test Pod 8 wires to the Test Point connector on the EECS373 I/O Board. TP0 should be connected to pin 31, TP1 to pin 30, TP2 to pin 29, TP3 to pin 28, TP4 to pin 27, TP5 to pin 26, TP6 to pin 25 and TP7 to pin 24.

Step 2. Setting up the Logic Analyzer

You will now perform a few basic LA measurements that you will need for the rest of this lab and future labs on your simple circuit. Power up your LA and wait for the Main System window to appear. Use the File Manager to select the "testpoints" configuration file in the eecs373 directory. Load the file by clicking the load button. On the Main System window, select the Workspace Icon and Double Click on the waveform icon. You will see a series of channels on left of the display labeled tp 0-15. These labels correspond to the individual probes on pod 8 numbered 0 through 15.

Step 3. Measuring with the Logic Analyzer

In normal operation, the LA continuously samples all of its channels as digital signals and stores the resulting binary values in an internal memory. Because the LA is sampling at a high frequency, it cannot display the signals as it records them. The internal memory is also finite, so only a small window of time is stored at any particular moment. In order to observe recorded signals, the LA must stop sampling. You can then use the GUI to browse through the signal values stored in the LA's memory.

Unfortunately, if you just randomly stop the LA and look at the recorded signals, the probability of seeing any interesting activity is basically zero. Instead, when you use a logic analyzer, you first tell it how to identify events you consider interesting. These interesting events are called *triggers*. When the LA sees a trigger condition, it will stop sampling and display the recorded signals at the point of the trigger. In your labs this semester, you will primarily use three simple triggers:

1. a rising edge on a particular signal, i.e., a transition from 0 to 1
2. a falling edge on a particular signal, i.e., a transition from 1 to 0, and
3. a rising or falling edge on one particular signal that occurs while one or more other signals have particular values.

We will try each of these conditions with this simple circuit.

The LA configuration file you loaded sets the LA to trigger on a falling edge of the TP0 signal. Since DIP switch 0 (DP0) is mapped to TP0, moving the switch from 1 to 0 should trigger the measurement. You must first start the sampling process by clicking the RUN button. The button will change to STOP with message window reading "Machine1: waiting in level 1". This means the LA is waiting for the trigger event.

Closing a DIP switch illuminates the respective BAR LED and produces a logic zero. Switch the DIP switch to determine which positions turn on and off the LED and produce logic zero and one respectively. Verify that producing a 1 to 0 transition on DP0 will trigger the analyzer. The DIP switches are arranged so that DP0 is to the extreme right.

By default, the LA dedicates half its memory to samples from before the trigger and half to samples after the trigger, placing the trigger event itself in the middle of its recorded time window. Use the scroll bar at the bottom of the window to view waveforms from before and after the trigger event.

Demonstration 1.1: Show your lab instructor that you have completed this step by triggering the LA with the appropriate input. Print the demonstration sheet at the end of this document and have your lab instructor initial. Demonstrations 1.1 through 1.4 can be shown together.

Step 4. Triggering the Logic Analyzer

Next you will learn to change the trigger to a rising edge condition. Click on the Workspace icon and double click on Machine1. These categories allow you to configure the LA measurements. The format category allows you to configure and organize the signals and the pods. We will not cover this capability in detail now. However, the format screen does

have one useful feature for this lab. Notice under each pod the logical state for each line is shown. Switch your dip switch to see them change. This can be a useful debug tool when tracing test signals. Select the Trigger category. Select Trigger Function. Notice that Trigger on an edge has been selected. Next select the Edge tab. Notice that the tp group is selected as the trigger source. The bar with 16 locations represents the trigger condition of each signal in the tp group. Double click the bar. Notice that position 0 has an arrow pointing down. This defines tp0 as the trigger with a falling edge. Each position can be set as indicated by the GUI. Use the GUI to change tp0 to a rising edge (arrow pointing up). Return to the waveform window and verify that the LA triggers on a rising edge.

Demonstration 1.2: Show your lab instructor that you have completed this step by triggering the LA with the appropriate input.

Next you should demonstrate that you can change the trigger from dip switch 0 to dip switch 1. Either edge condition is OK

Demonstration 1.3: Show your lab instructor that you have completed this step by triggering the LA with the appropriate input.

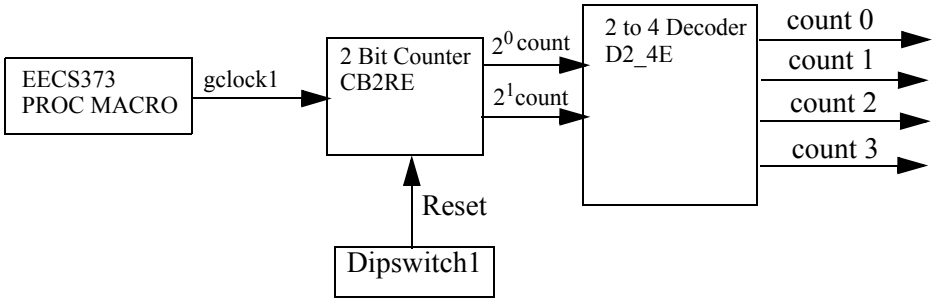
Finally, you will trigger the LA on an edge relative to a pattern. Under Trigger Functions tab, select Find edge AND pattern. You must select it and click the replace button. Next select the pattern tab. Next to the tp group select the format of the pattern to be binary. There are 16 locations corresponding to tp15-0 with tp0 located to the extreme right. Setting the value of each of the locations specifies a pattern. An X means either a 0 or a 1, a 0 means the value must be 0, and a 1 means the value must be 1. Set the locations so that dip switches d7-4 must be set to 1 and all other values are X. Select the edge tab. Dip switch 1 should still be set for the edge condition you select. Return to the waveform window and verify that the edge generated by switching dip switch 1 and the bit pattern 1111 on dip switches 7-4 must be present to trigger the LA.

Demonstration 1.4: Show your lab instructor that you have completed this step by triggering the LA with the appropriate input.

Part IV: Observing A Decoder with Logic Analyzer

In this section of the lab you will observe the operation of a simple decoder circuit. The circuit is provided as **T2** in \caen-mahogany\DFScourse\perm\eecs373\Lab1.

Create a separate project and implement the decoder. A block diagram of the decoder follows:



The test points are not connected in the schematic. To monitor the function of the address decoder, edit your schematic and assign test points to the clock that drives the binary counter, the binary counter outputs, the decoder outputs and the counter reset control.

Label the test points bus like the test circuit example to simplify LA connection, i.e., tp[0-14].

Remember, you will only need to use a buffer to connect your testpoints if you are connecting to a net that is already named. For example, the Reset signal is driven by dipswitch 1. Dipswitch 1 is a named net member of the dipswitch bus that needs to be connected to the testpoints bus. Consequently, a buffer is needed to connect the dipswitch 1 net to one of the nets of the testpoints bus.

Program the Xilinx FPGA with your address decoder. Configure the LA with the testpoints configuration. Connect the test pod connections to the respective test point connectors. Set the LA trigger to act on the falling edge of the binary counter reset signal. You should be able to see that the counter is held in reset while the reset signal is high and the counter begins to count after it is released from reset when the reset is low.

Demonstration 2.1: Show your lab instructor that you have completed this step by triggering the LA with the appropriate input. Demonstrations 2.1 and 2.2 can be shown together.

Notice that the binary counter counts on every rising edge of the input clock. The binary output is stable during the falling edge of the clock. In future labs, you will need to trigger the LA on address decodes to verify decoder designs. Trigger the LA on the falling edge of the clock and count 1. That is, use the clock as the trigger and count 1 as the pattern. You may either use the counter outputs or the decoder output for the pattern.

Demonstration 2.2: Show your lab instructor that you have completed this step by triggering the LA with the appropriate input.

Appendix A: Sharing Class Directory Space

To share eecs373 class directory space use the following procedure.

1. Select or highlight the class directory you want to share and right click.
2. Select Properties.
3. Select the security tab.
4. Click the Add button.
5. In the bottom window, type the uniquename of the person you wish to share.
6. The system will search for the uniquename and will return to the previous window with the added name if you were successful.
7. Select the permission you wish to grant this user in the permissions window.
8. Click apply
9. Click OK to exit.

Lab 1 Demonstration Sheet

Print this page and present it to your lab instructor when demonstrating the various lab sections. Turn this sheet in with your post lab or when your in lab demonstration is due. You are required to turn in only one demonstration sheet per group.

List Partners Names

_____

_____

_____

Part IV Demonstrations, Triggering the Logic Analyzer

D1.1 through D1.4 demonstrations may be shown to your lab instructor together.

D1.1 Verify the LA is triggered on the falling edge with dipswitch 0.

Lab instructors initials:

D1.2 Verify the LA is triggered on the rising edge with dipswitch 0.

Lab instructors initials:

D1.3 Verify the LA is triggered on the either edge with dipswitch 1.

Lab instructors initials:

D1.4 Verify the LA is triggered on either edge of dispwitch 1 with dipswitches D7-D4 set to logical 1. Try to trigger without D7-D4 set to 1 to verify LA is triggering on the 1111 pattern.

Lab instructors initials:

Part V Demonstrations, Observing a Decoder with the Logic Analyzer

D2.1 through D2.2 may be demonstrated together

D2.1 Verify the LA is triggered on the falling edge of the binary counter reset signal. The clock, binary counter output and decoder outputs should be identified by the group.

Lab instructors initials:

D2.2 Verify that the LA is triggered on the decode 1 and the falling edge of the clock.

Lab instructors initials: