UNIVERSITY OF CALIFORNIA AT BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Lab 5
Logic Analyzers

1.0 Motivation

In the last lab, you had a chance to become familiar with the basic debugging techniques which will allow you to finish your project in this course. This week you are being asked to work with more sophisticated hardware debugging tools: the HP/Agilent Logic Analyzers and Xilinx's software logic analyzer: ChipScope.

Logic analyzers are the most powerful tool available for debugging digital circuits, they can potentially show every signal in your circuit on every clock cycle. As you might expect this power comes with a price: using a logic analyzer requires practice, therefore it is the goal of this lab to help you gain experience with these powerful tools before you need to work on your final project.

2.0 Introduction

In this lab you will be working with two tools, the first of which is the bench logic analyzer from HP/Agilent. These bench analyzers use specialized probes and expensive hardware to capture digital signals from your circuit. The main virtue of these analyzers is their ability to zoom in and out in time, so as to see the operation of your circuit over long time periods (10msec). As good as there are however these logic analyzers are expensive and have some drawbacks, hence the second tool you will use.

You will debug a broken counter circuit in the FPGA using a software-based logic analyzer called ChipScope.

The bench logic analyzers in the lab only have 16 binary channels, which is often insufficient for debugging more complicated designs. In addition, it is sometimes hard to set the correct triggering modes and get the data you need from these logic analyzers.

Thankfully, Xilinx offers ChipScope, which will allow you to integrate a logic analyzer with your design and then control it from the computer. With ChipScope you can examine significantly more than 16 signals, use more powerful triggering modes, and even save the data for later analysis.

3.0 Prelab

Please make sure to complete the prelab before you attend your lab section. **You will not be able to finish this lab in 3hrs otherwise!**

1. **Read this handout thoroughly**. Pay particular attention to section **4.0 Lab Procedure** as it describes what you will be doing in detail.
2. **Examine the Verilog** provided for this weeks lab.

   a. You should become intimately familiar with the **FPGA_TOP2.v** files as **you will need to debug with it.**
   b. Watch the ChipScope tutorial, examine the example verilog
      i. http://www-inst.eecs.berkeley.edu/~cs150/sp05/Documents.htm#Tutorials
3. You will need the **entire 3hr lab**!
   a. You will need to test and debug both your verilog and ours.
   b. **You will be asked to use the logic analyzer for the first time.**
   c. **You will be asked to use the ChipScope for the first time.**
      i. It's easy to use, but you will need to play with it to learn it

4.0 Lab Procedure

4.1 Bench Analyzer

In this part of the lab you will be **examining an FSM**, similar to Part 3 of Lab #5. However this time you will not have the luxury of being able to single step through the various states. **This FSM has no enable signal**, and will therefore take an input on every clock cycle. At 27MHz, this means that it could potentially go through 27 million states per second, much too quickly for you to observe on the LEDs, let alone record.

Shown in Figure 5 below is a **high level block diagram of the test harness** we have built for you in **FPGA_TOP2**. Your job is to connect the **logic analyzer** at your station to the **pins listed in parentheses**, set up a **pattern of 8 inputs** to be fed to the FSM and then push **Reset** (SW1). This will **reset the FSM** and then **feed it those 8 inputs in a continuous loop**, allowing you to **trace through the various FSM states** on the logic analyzer.

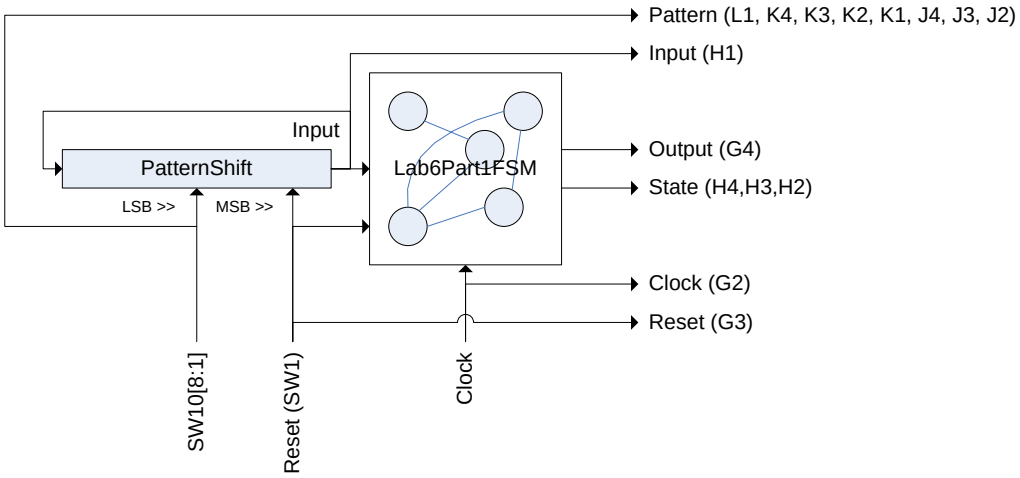

Figure 1: Part1 Test Harness

We have provided you with the code for this test harness and the shifter, please **read the code thoroughly** as it will make your job during lab significantly easier. We have again given you only a black box for the FSM. **For more information about working with black boxes, see section 4.2 in Lab #5**.

You job is to derive a complete bubble-and-arc diagram for this FSM. Keep in mind that one input pattern may show you most of the FSM. **This FSM uses all 8 states**, you must **find and label all 16 arcs as well as the outputs**.

PLEASE READ SECTION 5.0 HP54645D Mixed Signal Oscilloscope (Logic Analyzer) ***CAREFULLY! TAKE SPECIAL NOTE OF*** 5.1.8 Digital**,** 5.1.9 Trigger ***AND*** 5.1.10 Storage ***NOT TO MENTION*** 5.2 Triggering**.**

4.2 ChipScope Integrated Logic Analyzer

For this part of the lab we have provided you will two counters: **Counter** and **BrokenCounter**. They should be functionally identical, however as you might guess **BrokenCounter** is very buggy. It is your job to figure out what the **BrokenCounter** is actually doing. The easiest way to do this will be to instantiate both counters in **FPGA_TOP2+.V** and then compare their output. When their output is not equal you will want to trigger the ChipScope logic analyzer.

You must fill in **FPGA_TOP2+** with the testing harness needed to determine what **BrokenCounter** is actually doing. Your goal for this part of the lab is to tell us what the problem with **BrokenCounter** is, using only a verilog test harness of your own design and ChipScope. We recommend that you work with the ChipScope example and walkthrough available on the documents page of the CS150 website at http://www-inst.eecs.berkeley.edu/~cs150/sp05/Documents.htm#Tutorials.

4.2.1 Working with ChipScope

ChipScope is an embedded, software based logic analyzer. By inserting an "integrated controller core" (**icon**) and an "integrated logic analyzer" (**ila**) into your design and connecting them properly, you can monitor any or all of the signals in your design. ChipScope provides you with a convenient software based interface for controlling the "integrated logic analyzer," including setting the triggering options and viewing the waveforms.

There are six main steps to using ChipScope, as detailed below.

1. Generate an "integrated controller core" or **icon**
2. Generate one or maybe more "integrated logic analyzers" or **ila**s
3. Connect the **ila**s to the **icon** and make all of these modules part of your design.
4. Synthesize, and implement your design (including the **icon** and **ila**) as normal.
5. Program the CaLinx board
6. Run the ChipScope software to access and use the **ila**s (the ChipScope software requires the **icon** to gain access to the **ila**s)

For a more detailed ChipScope tutorial, please refer to the documents page of the website (http://www-inst.eecs.berkeley.edu/~cs150/sp05/Documents.htm#Tutorials) where you will find both a ChipScope tutorial document and an accompanying video.

5.0 HP54645D Mixed Signal Oscilloscope (Logic Analyzer)

Figure 6: HP54645D Mixed Signal Oscilloscope

Shown in Figure 6 above, is the HP54645D Mixed Signal Oscilloscope, which we will generally refer to as simply "the oscilloscope" or "the logic analyzer." It is in fact a combination of these instruments, both of which are design to graph waveforms (analog and digital respectively) over time. In this class we will almost exclusively use the logic analyzer portion of these devices, since this is a digital systems class.

Using the logic analyzer is significantly less complicated than it appears and it is an invaluable debugging tool. The logic analyzer allows us to examine signals over time at various scales, you can zoom in to see events on different clock cycles, or zoom out to see how a signal behaves over the course of seconds.

5.1 Front Panel Controls

This section will explain at least a little bit about all of the buttons and controls on the front panel of the logic analyzer.

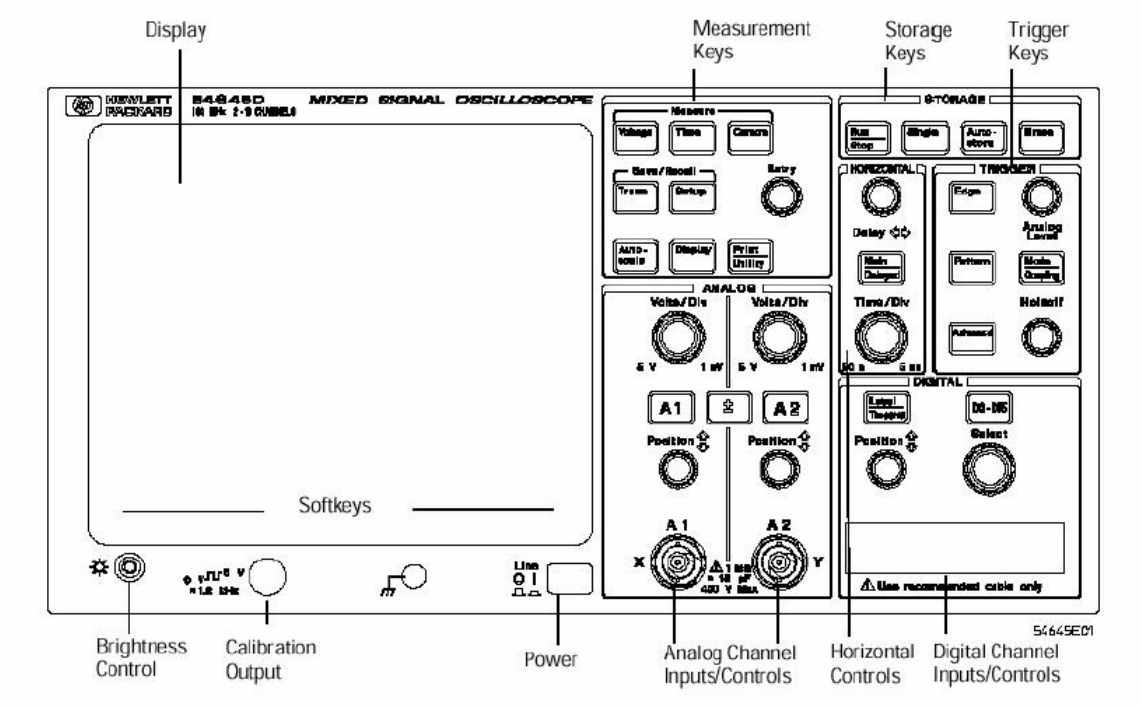

Figure 7: HP54645D Front Panel Controls

5.1.1 Display & Brightness Control

Of course this is the screen, where you will see your waveforms and other such information. At the bottom left of the screen is a brightness control. If your screen appears blank, try the brightness.

Notice the row of softkeys across the bottom of the screen.

5.1.2 Softkeys

In order to provide more advanced controls without cluttering the front panel, there are six softkeys, whose meaning changes as you navigate the controls menus or change operating modes. If we mention a button that does not appear on the front panel, check the softkeys.

5.1.3 Calibration Output

This is used to calibrate the analog oscilloscope probes. If you are in doubt as to whether the probe is behaving correctly, hook it to the calibration output and hit **Autoscale**. You should see a nice clean 0-5V square wave at ~1.2kHz.

5.1.4 Autoscale

Located in the **Measurement Keys** section of the front panel, this is one of the most used buttons. It will cause the oscilloscope/logic analyzer to automatically search for signals on its inputs. When it finds signals it will attempt to place them in the middle of the screen and set the triggering to that they are stable and ready for use.

If you lose the signal, **Autoscale** may help.

5.1.5 Measurement Controls

Consists of three buttons; **Voltage**, **Time**, and **Cursors**. **Voltage** and **Time** will display options for types of measurements (V_{avg}, V_{p-p}, Freq, etc.) on the softkeys. When selected, the measurements will be displayed at bottom of screen. The **Source** softkey can be used to select the correct analog source channel.

The **Cursors** button is used to display or clear time and voltage measurement cursors which are selected with softkeys and moved with the **Entry** knob. Note that you can have the time cursors print out the value (in Hex, Binary or Decimal) of the 16 digital channels.

5.1.6 Analog Controls

The **Volts/Div** knob can be used to change the vertical (voltage) scaling of the two analog channels. Notice that there are two knobs: the scaling for each of the two channels are separate (1 division may be 1V on A1 and 2V on A2).

The **A1** and **A2** keys will turn the respective analog channel on or off while bringing up a softkey menu allowing you to select AC or DC coupling and Noise Filtering. The ± button will allow you to see sum and difference waveforms.

5.1.7 Horizontal

The **Delay** knob will allow you to scroll the screen left or right.

Main Delayed: Horizontal mode should be set as normal with softkeys, the **Time Ref** softkey controls what reference to use when zooming in with **Time/Div** knob. **Vernier** is used to minimize time steps in **Time/Div** knob.

Time/Div changes the time per grid block. The time/div scaling is displayed at the top of the screen.

5.1.8 Digital

The **Label/Threshold** button allows you to label any of the 16 probe leads of the logic analyzer. The **threshold** menu and softkeys let you choose what type of logic you are using, MOS or TTL should work fine.

The **D0-D15** button allows you to select which logic analyzer probes will be displayed. Individual signals can be selected with **Select** knob and turned on or off with the softkeys.

The **Position** knob will let you reorder the signal on the display. Notice that this will affect the value readout produced by the time cursors (see 5.1.5 Measurement Controls).

5.1.9 Trigger

The **Edge** button will select edge triggering mode, where the scope will wait for a rising or falling edge (selectable with softkeys) on the channel you select using the softkeys.

The **Pattern** button will let you modify the pattern triggering mode pattern, where the scope will trigger on a pattern of High, Low, Rising, Falling or Don't Cares on the various inputs. Use the **Select** knob and softkeys to set the pattern.

The **Analog Level** knob will let you set the analog voltage at which to trigger (applies to the analog signals only). The **Holdoff** knob will allow you to set the amount of time before the screen is redrawn; you will want to set this to the minimum.

The **Mode/Coupling** button will bring up a softkey menu of triggering modes: **Auto Level**, **Auto** and **Normal**. While **Auto** may be useful for analog work, you will always want **Normal** for using the logic analyzer.

The **Advanced** button brings up a softkey menu of the various triggering modes.

5.1.10 Storage

The **Run/Stop** button starts and stops the refresh on the display. This is useful when you want to freeze the screen to examine it more closely.

The **Single** button can be used to set the scope to single mode, where it will trigger once, remember the data and then freeze the screen. This is perfect for capturing a one time event, like something that happens at reset.

Erase will clear the screen, measurement should be restarted with the **Run/Stop** button.

5.2 Triggering

Triggering the logic analyzer properly is key to being able to actually see the waveforms you are interested in. We recommend that you become intimately familiar with the various triggering modes, but perhaps the most useful for digital work is pattern mode. This section is a brief tutorial on how to set up and use pattern triggering mode.

1. Make sure the logic analyzer is on
2. Make sure the **channels you are interested in** are showing
    a. The **A1** and **A2** buttons can be used to enable or disable the analog channels.
    b. The **D0-D15** button and **Select** knob will let you select specific digital signals. You can then use the **softkeys** to enable or disable them.
3. Switch the logic analyzer to **Normal triggering mode**.
    a. Press **Mode/Coupling** in the **Trigger Box**
    b. Select **Normal** under **Trigger Mode** on the softkey menu
4. Switch the logic analyzer to **Pattern advanced triggering mode**.
    a. Press **Advanced** in the **Trigger Box**
    b. Select **Pattern** under **Advanced Trigger Mode** on the softkey menu
5. Set up a triggering pattern
    a. Press **Pattern** in the **Trigger Box**
    b. Use the **Select** knob to select a signal
        i. Select the value that signal should have
        ii. Note that each signal is compared against its bit in the pattern, when all 18 (16 digital, 2 analog) channels match, the scope will trigger
        iii. L: Logic Low
        iv. H: Logic High
        v. X: Don't Care (Ignore this signal for the pattern)

vi. ↑: Rising Edge
vii. ↓: Falling Edge

6. Set the **Time/Div**
    a. Play with the **Time/Div** knob until the time per division is approximately what you need.
    b. Remember a 27MHz clock cycle is 37ns from rising edge to rising edge.
7. Start the logic analyzer running
    a. To trigger once and save the waveforms press **Single**
        i. If you want to trigger again you will need to press **Single** again
    b. To trigger every time the pattern is matched press **Run/Stop**

5.0 Lab5 Checkoff

Name: _____ SID: _____
Name: _____ SID: _____
Section: _____

I Part I: State Transition Diagram _____ (50%)
1 Draw the diagram on the back of this sheet
II Part II: Describe the output of BrokenCounter below _____ (50%)

III Hours Spent: _____

IV Total: _____
V TA: _____

10/2/2005	David Lin	Changed to Lab 5
RevA – 1/30/2005	Greg Gibeling	Created a new lab Based on the old Fall 2004 Lab4 and Spring 2004 Lab5