USB Logic Analyzer

System Block Diagram

Shom Bandopadhaya
Advisor: Dr. James H. Irwin

November 8, 2005

Senior Capstone Project
Bradley University
Department of Electrical and Computer Engineering

Introduction

A logic analyzer displays logic level of digital signals. A logic analyzer differs from an oscilloscope which is a more powerful instrument but typically only has a few channels. This project seeks to continue work on a digital logic analyzer which has sixteen channels. The logic analyzer will display four logic levels: low, high, tri-state and indeterminate. The sixteen channels are sampled by an external conditioning hardware called a POD. The POD consists of an Opal Kelly XEM 3001 FPGA board and some external hardware. The POD interfaces to the computer using Universal Serial Bus [USB] protocol. The graphical user interface [GUI] on the computer will display one of the four possible logic levels sampled by the POD. The overall system block diagram can be seen in Figure 1: Overall system block diagram for USB Logic Analyzer.

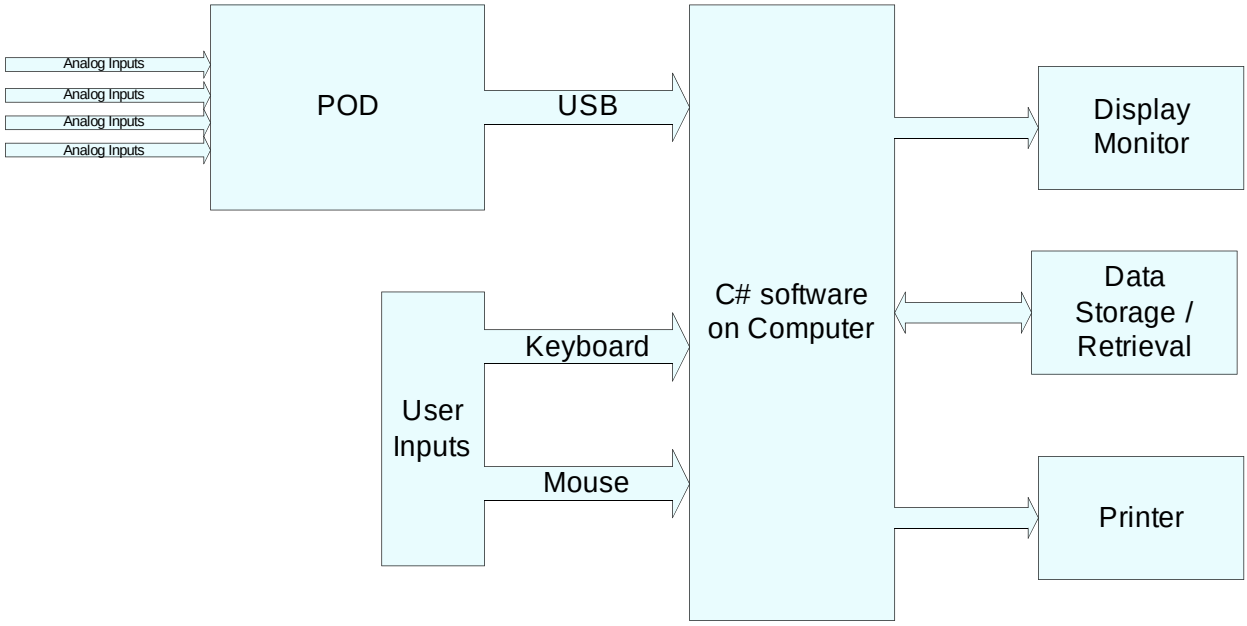

Figure 1: Overall system block diagram for USB Logic Analyzer

Inputs

Analog inputs:	These are the analog voltages on the probes of the POD. The POD handles CMOS or TTL levels.
User inputs:	The user interacts with the software GUI using the keyboard and mouse, and is able to select options, change display parameters, and access additional features like zoom, scroll and save.

Outputs

Display Monitor:	Displays the GUI. The user can see the current logic levels, triggering points and also observe the effects of their interactions with the GUI.
Data Storage/Retrieval:	Saves the captured data frame to disk. The GUI can also open and display previously saved captures.
Printer:	Prints the current screen.

Operation Modes

Option selection:	Allows the user to select from the available options, such as logic type (TTL or CMOS), display window range, etc.
Single Capture:	Captures and displays information from the time the capture was started till the time software buffer was filled.
Continuous Capture:	Continuously captures information into the buffer and displays the current buffer, so the logic levels are shown at “pseudo real-time”.

Goals

- Primary
    - New display engine
    - DLL interface for USB
    - Efficient data processing
    - Print/Save data capture
    - Continuous Update (Pseudo Real Time)
- Secondary
    - Remote Viewing
    - Reverse Assembly code decoding

Software

Conditioned Signals: Signals sent from the POD to the PC with all sampled data in packet form USB interface.

Keyboard/Mouse: Commands entered using keyboard and mouse to setup trigger conditions, change display format, save the waveform, setup, and hard copies.

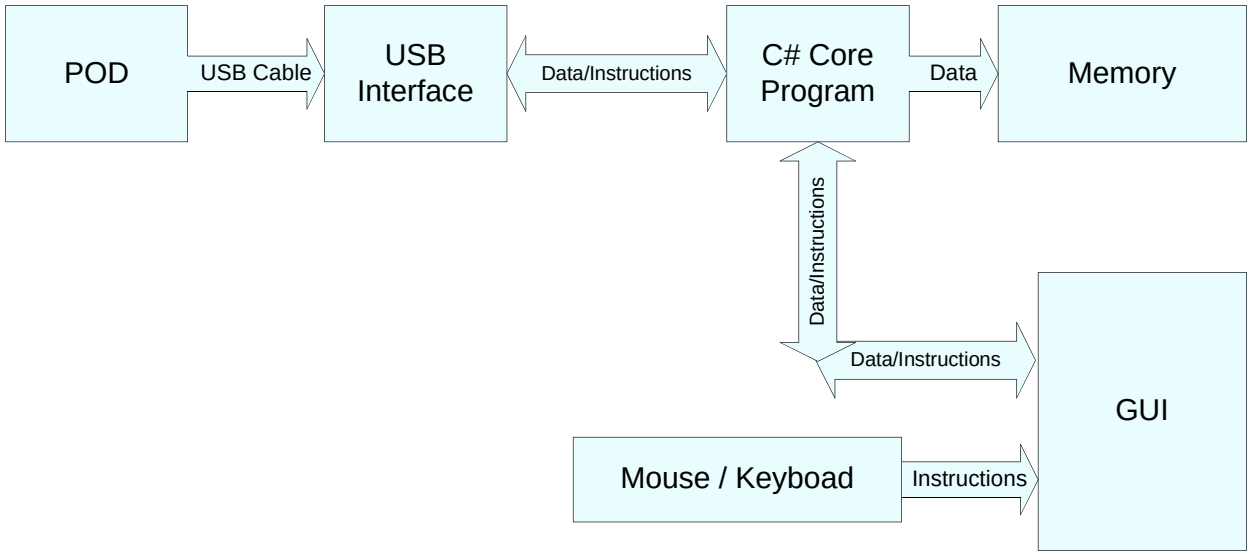

Figure 4.1 Software Subsystem Block Diagram

There are three major sub-blocks, the USB Interface, the C# Core Program, and the GUI as can be seen in Figure 4.1: Software Subsystem Block Diagram.

USB Interface

The USB interface uses the Opal Kelly provided library to connect to the FPGA using a USB 2.0 connection. It acquires data when requested by the C# core program and transmits the data back to the C# core program through the USB buffer. Figure 5.1: USB Interface Tasks shows the process through which it is accomplished.

C# Core Program

The C# Core Program receives the instructions via the GUI, contacts the USB interface to start Data Acquisition, collects the data and then saves it to memory and sends it to the GUI to be displayed. Figure 5.2: C# Core Program Tasks describes the process.

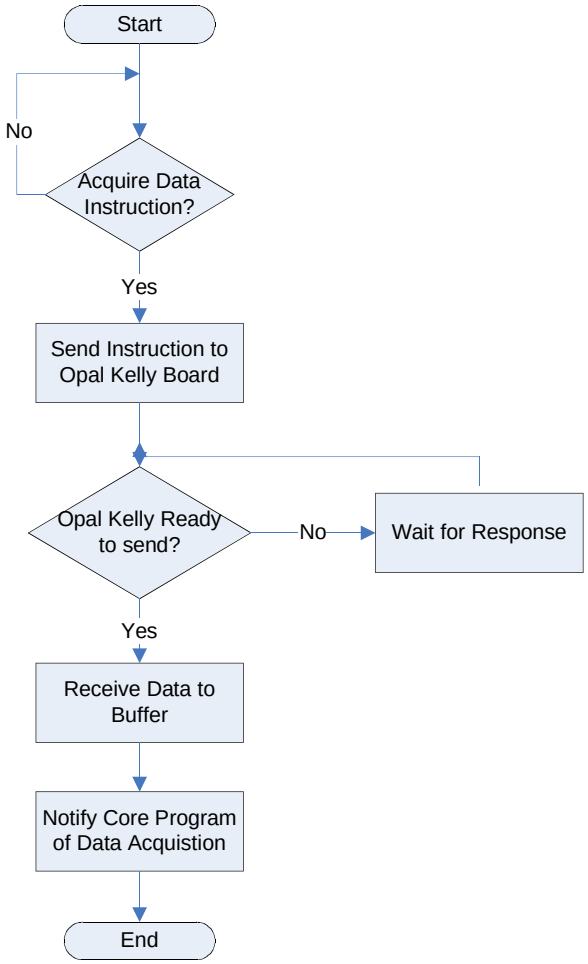

Figure 5.1 USB Interface Tasks

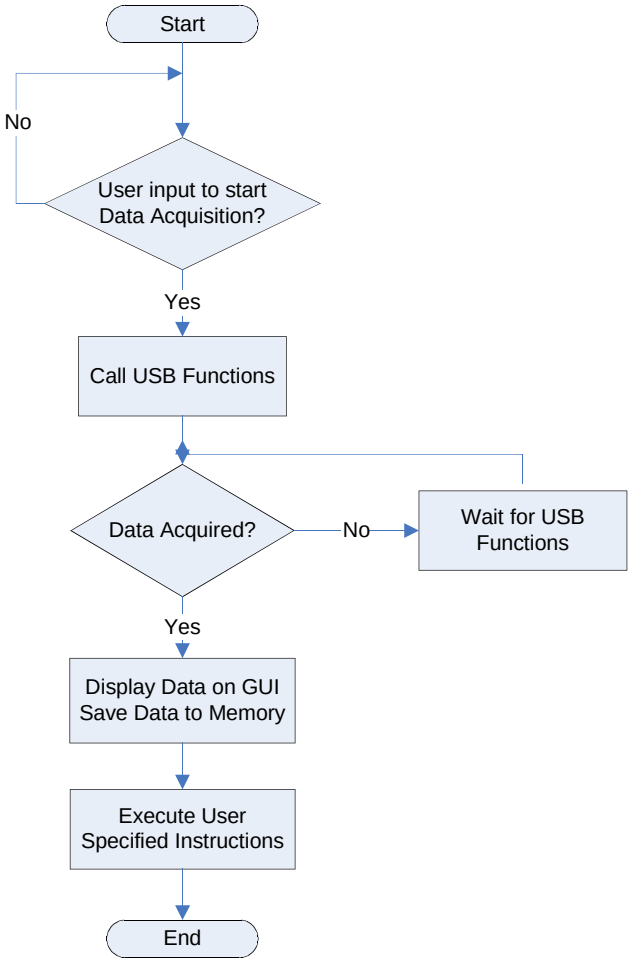

Figure 5.2 C# Core Program Tasks

GUI

The GUI interfaces the user to the C# core program. It receives instructions via the keyboard and mouse and displays the information on the screen. Once it receives any instructions it contacts the Core C# Program to perform those tasks. It idles till it receives the next set of instructions. If the user decides to terminate the GUI it terminates the whole software. Figure 6.1: GUI Tasks shows this process.

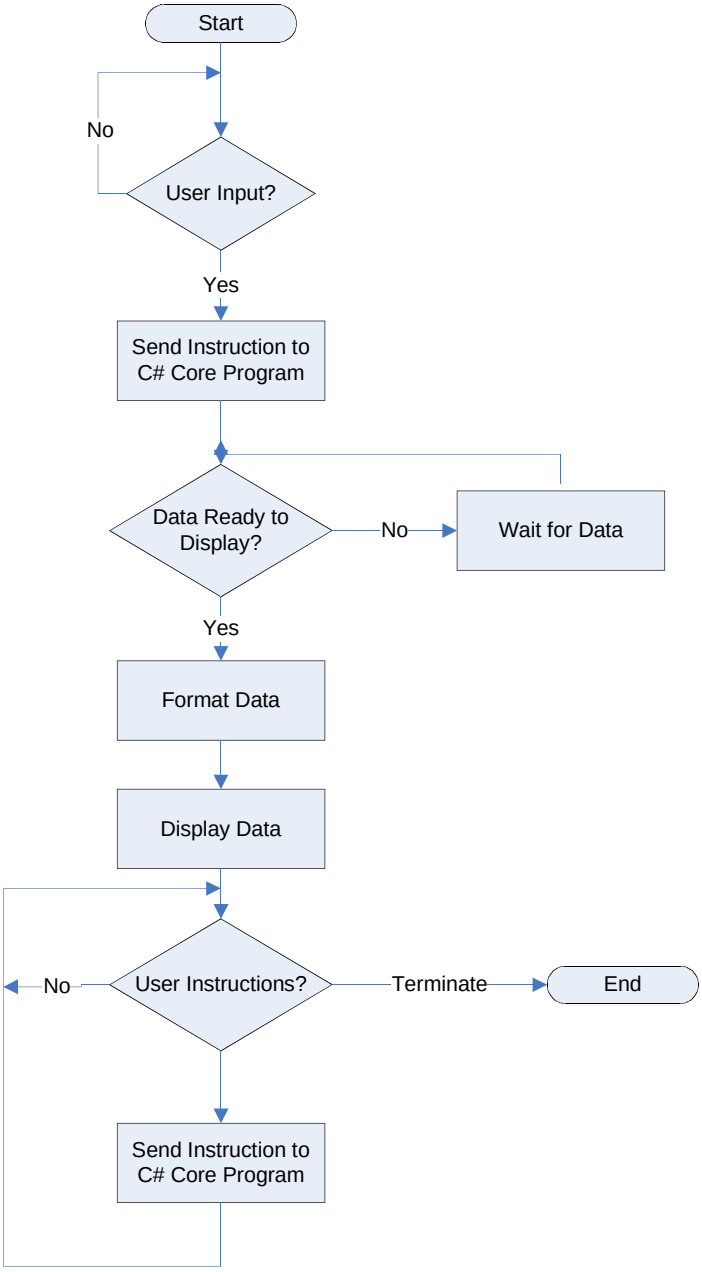

Figure 6.1 GUI Tasks