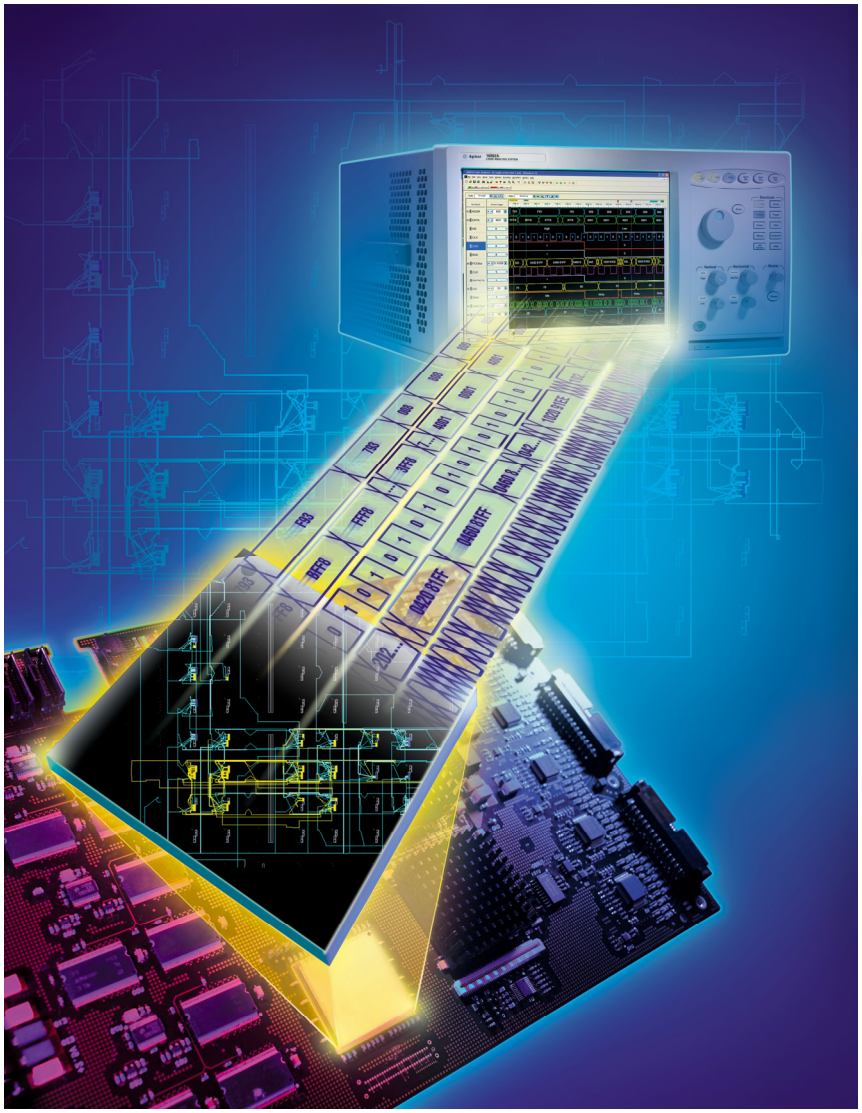


1 FPGA Overview/Background

FPGAs are pervasive with more than 80K design start per year (10X number of ASIC design starts). In order to differentiate their products and incur lower development risk, design teams continue to implement bigger and faster FPGAs in nearly every design. Communications, industrial, automotive, aero-defense, wireless, medical, education, and T&M all make extensive use of FPGAs. Even computer servers, which have long relied on ASIC technology, have arrays of FPGAs that allow greater design flexibility as well as field upgrades. In industries where ASIC is king such as automotive, consumer, and semiconductors, FPGAs are often used for prototyping ASICs.

Because FPGAs are reprogrammable, designers tend to quickly move from simulation to real-world testing where a few seconds in-circuit can reveal what would have taken weeks to months to simulate. Scopes and logic analyzers are the primary debug tools for HW engineers using FPGAs. MSOs have great and unique FPGA debug capability for developers using FPGAs who historically used scopes for debug. With their ability to measure signals broadly at numerous points across the analog, digital, and serial domains, a good MSO is invaluable when searching for an elusive problem in a complex system of hardware and software. Logic analyzers excel at quickly finding functional problems and have extensive digital triggering capabilities.

FPGAs are typically the last HW block to be implemented. While not sure what kinds of problems they will run into, seasoned FPGA designers design for debug. The approach they typically use is to take advantage of the programmability of the FPGA to route internal nodes to a small number (typically 8-32) of physical pins.

Limitations & Pain

While this is a very useful approach, it does have some serious limitations. First, since pins on the FPGA are typically an expensive resource, there are a relatively small number available, limiting the visibility of internal nodes to the same small number of signals (i.e. one pin is required for each internal signal to be probed). When different internal signals need to be accessed, a recompile of the design may be required. This can be very time consuming and often changes the timing of the FPGA, and so is definitely something to be avoided.

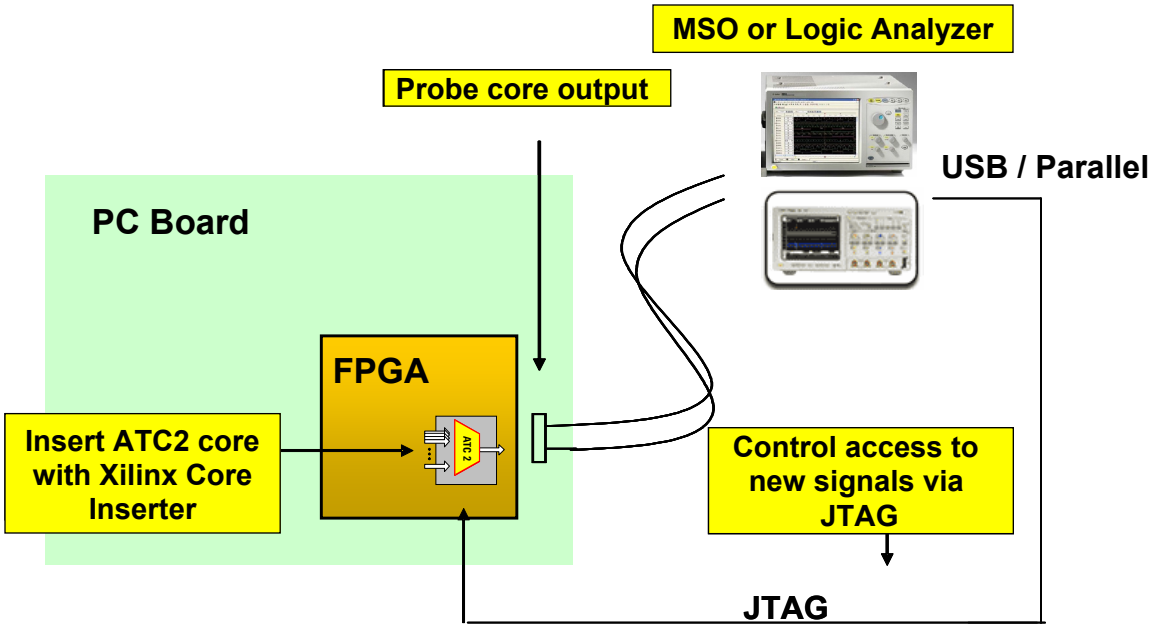
Finally, the process required to map the node names from the FPGA design to the LA/MSO labels is completely manual, and is tedious and error-prone at best. When changes are made, the need to manually update these label names to match the new configuration of the measurement (potentially very useful and important) takes additional time and is a potential source of confusing errors. Since there often may be many iterations, this becomes even more of a problem.

2 FPGA Dynamic Probe Overview

FPGA dynamic probe enables incremental in-chip probing without stopping the target system, without modifying the design to route out a new set of signals, and keeps timing constant as probe points are changed. FPGA dynamic probe also greatly simplifies the task of setting up the LA/MSO for the first, and successive measurements.

Making use of FPGA Dynamic Probe starts with an assessment of the internal nodes of the FPGA that may benefit from logic analysis access, and the type of measurement (state analysis or timing analysis) that needs to be made. The user then selects the appropriate core (from a list of options developed incorporated into the Xilinx and Altera FPGA development systems, and inserts it into the design. The inputs to the core are the potential internal nodes to be probed by the LA/MSO digital channels, and the outputs are routed to a set number of physical pins dedicated to debug. Incremental work can associated with the additional of an ATC2 core typically takes less than 30 minutes.

The cores can be thought of as MUXes that are controlled via the JTAG connection to the FPGA offered through the standard Xilinx or Altera JTAG cables used to download and program the FPGA. This whole system is controlled via Agilent's FPGA dynamic probe software.



3 Xilinx Spartan-3E FPGA Demo Setup Overview

3.1 To run the Xilinx Spartan-3E FPGA demo, first, you have to install the '**B4655A Xilinx FPGA Dynamic Probe'** application software.

3.1 Install the USB driver to Logic Analyzer by following the '**4. USB JTAG Driver Installation Guide**'.

3.2 Now, you can start to demonstrate the Xilinx Spartan-3E FPGA demo by follow the demo guide either using a Flying lead set (see **6. Xilinx Spartan-3E FPGA Demo Guide**).

(Note: Downloadable at: http://field.cos.agilent.com/apps/fpga-debug/sales/#logic. Since 16800 and 16900 logic analyzers do not have a CD-ROM, you will either need an external USB CD-ROM, or you will need to transfer files from the provided CDs to a USB thumb drive and then from the thumb drive to the logic analyzer)

Demo Requirements:

Hardware

1. 1680, 1690, 16800, 16900 Series software version 03.50 or greater
2. Xilinx Spartan-3E demo board
3. USB cable (Standard Type A to Type B USB cable)

Software

1. B4655A Xilinx FPGA dynamic probe application SW installed on Logic Analyzer
2. Xilinx .bit configuration file and .cdc naming files
   a. atc2_state_8mA.bit
   b. atc2_state.cdc

Probing

1. Flying Leads

4 USB (JTAG) Driver Installation Guide

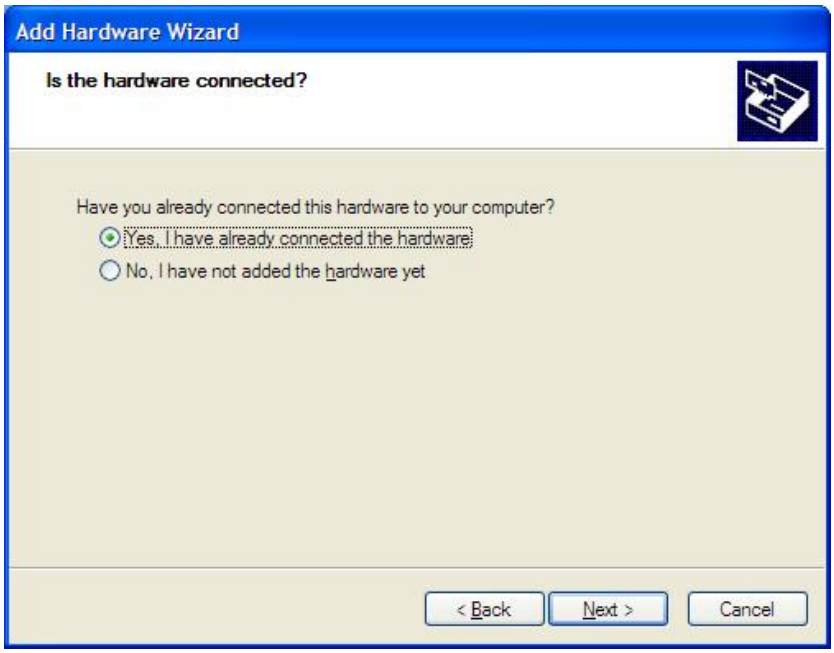
Remarks:

i. **B4655A Xilinx FPGA dynamic probe application SW must be first installed on the system.**
ii. **Make sure the USB (JTAG) cable is connected between demo board and Logic Analyzer with demo board turned on.**
iii. **If you connect the board via USB and the red LED illuminates and stays constant after the driver has been properly installed, it means the firmware driver on the USB Platform cable hardware is being updated (this can take quite some time).**

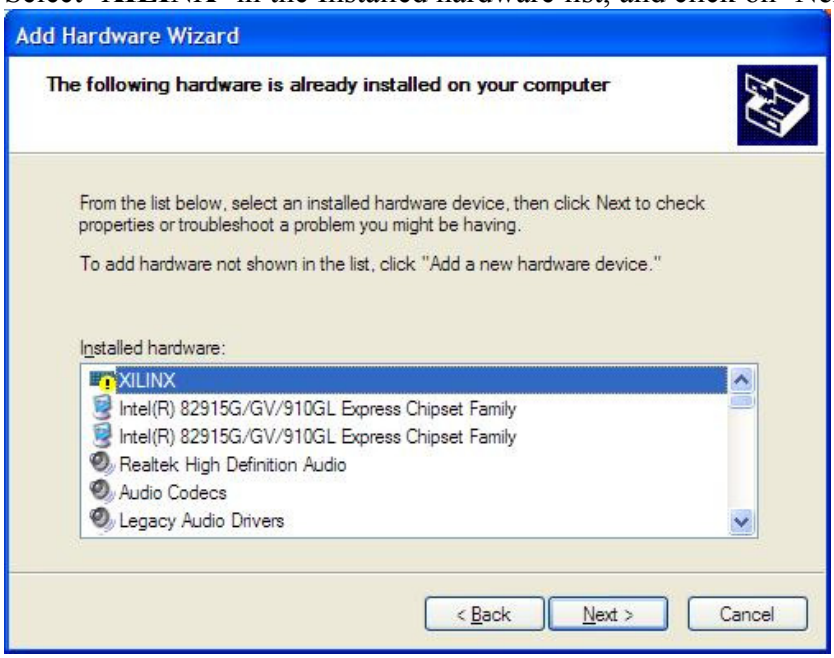
4.1 If this is the first attempt at install, you will be prompted with the Found New Hardware Wizard or else click on the Start menu, and then select '**Control Panel**'.

4.2 Double-click the '**Add Hardware**' icon to start the Add Hardware Wizard and click on '**Next**' to continue.

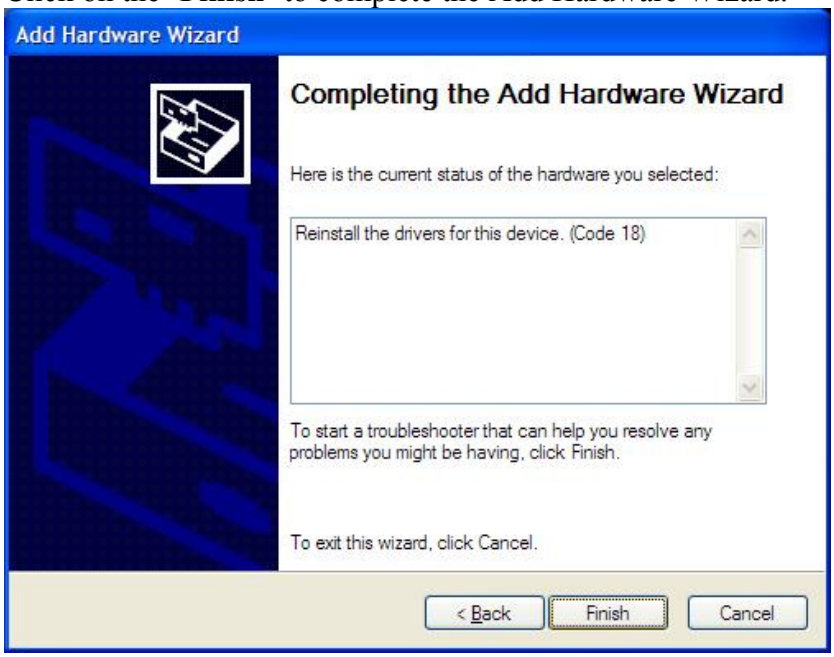
4.3 Select '**Yes, I have already connected the hardware**' and click on '**Next**' to continue.



4.4 Select '**XILINX**' in the Installed hardware list, and click on '**Next**' to continue.



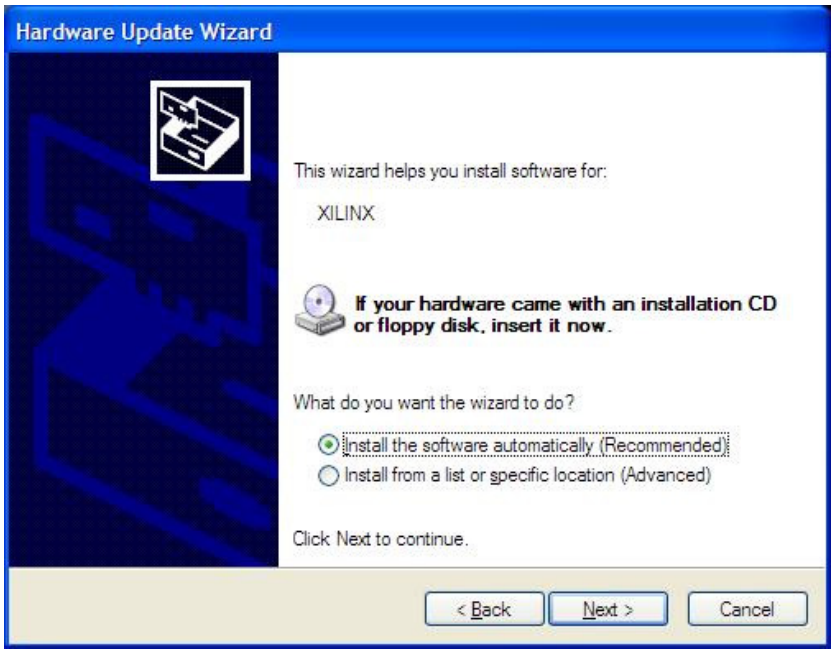
4.5 Click on the '**Finish**' to complete the Add Hardware Wizard.



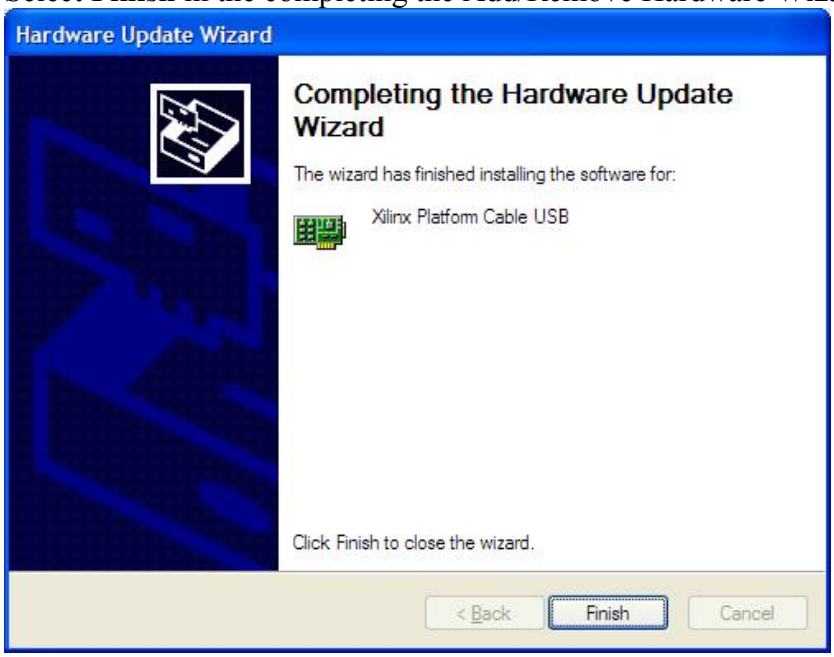
4.6 The '**Hardware Update Wizard**' will automatically show on the screen as per the figure shown below. Select '**No, not this time**' and click on '**Next**' to continue.

Note: Don't search for new software on the web, you shouldn't need to because it can find the drivers locally.

4.7 Select 'Install the software automatically (Recommended)' and click on 'Next' to continue.

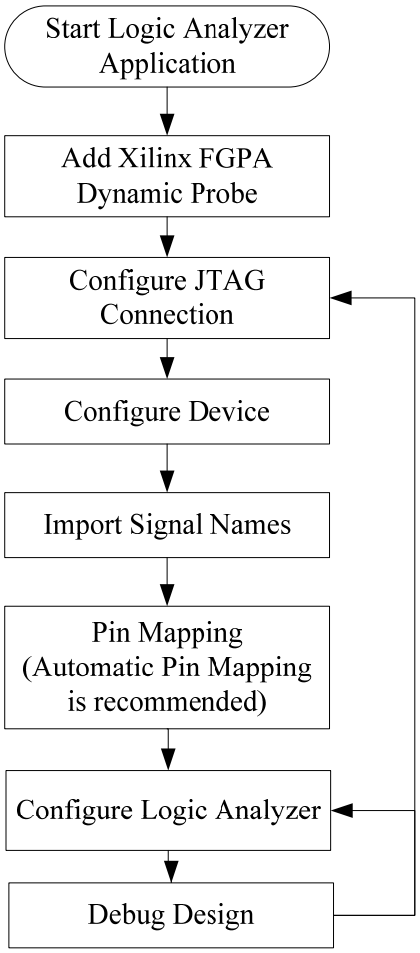


4.8 Select **Finish** in the completing the Add/Remove Hardware Wizard window.



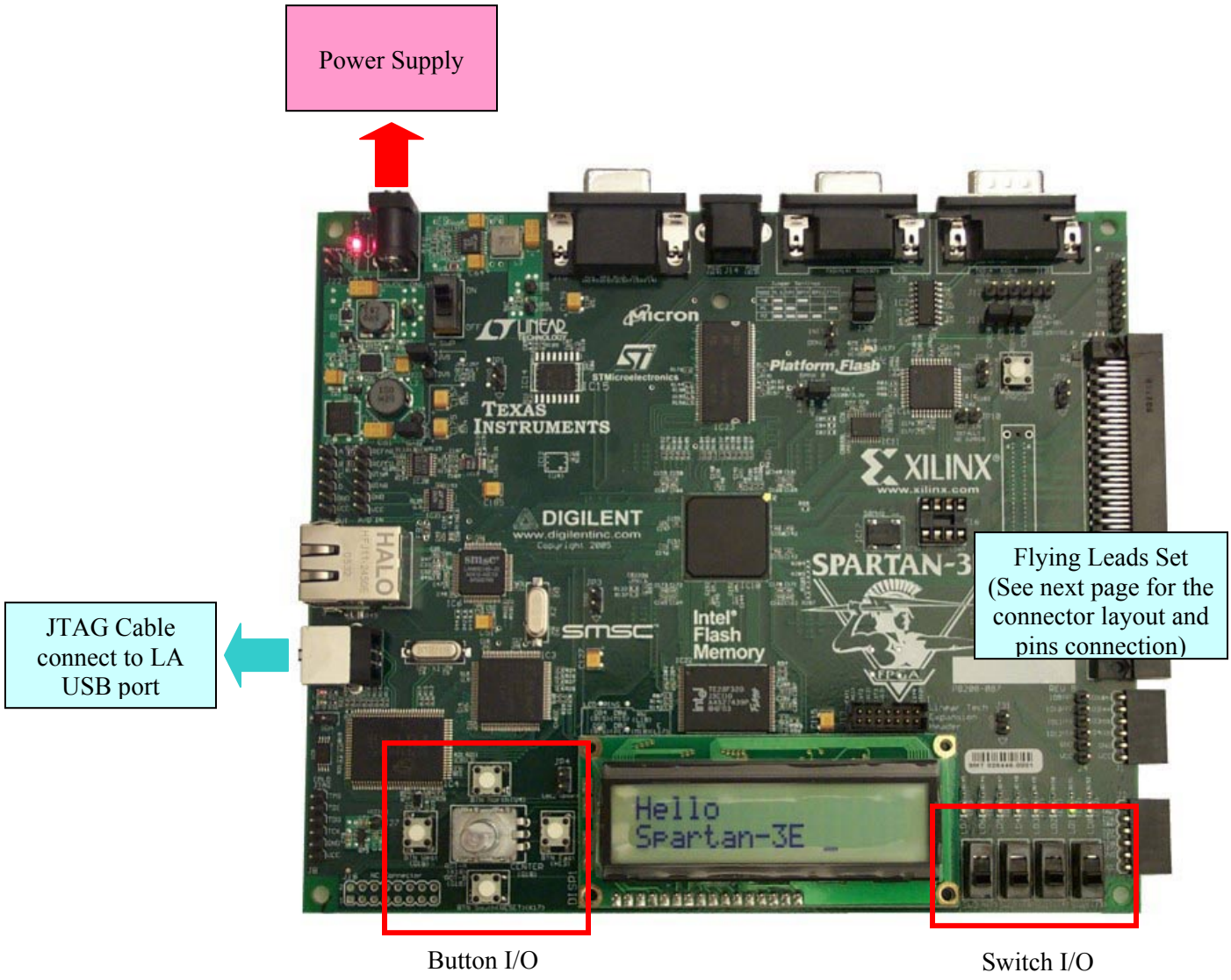
4.9 Reboot the system.

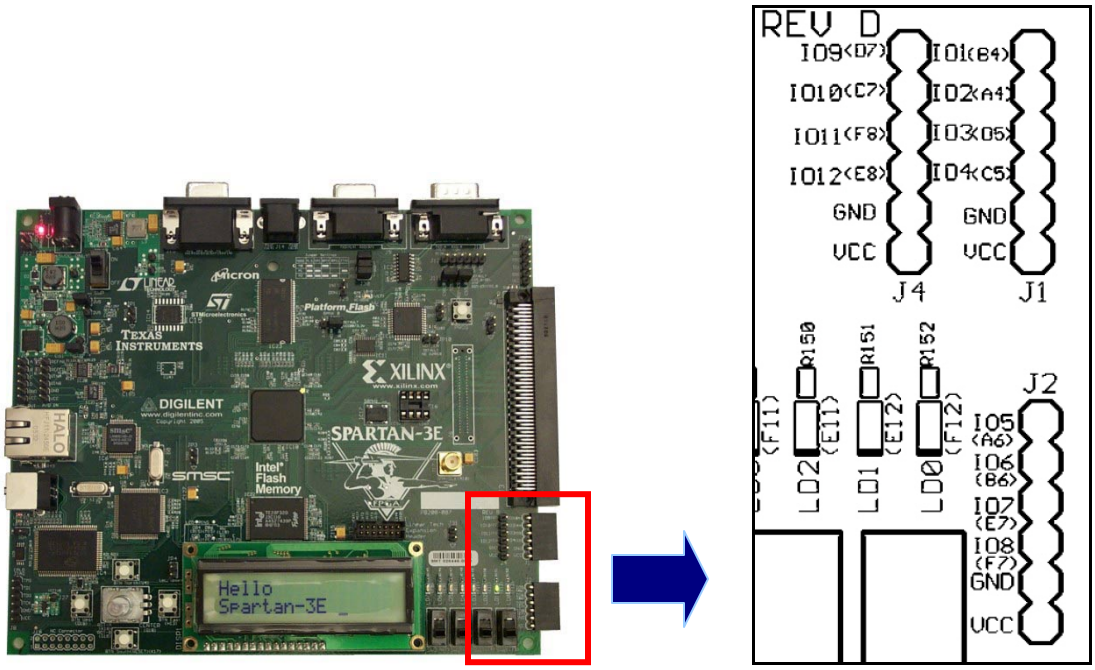
5 Xilinx FPGA debugging Process flow



6 Xilinx Spartan-3E FPGA Demo Guide

6.1 Connect the connections according to the figure shown below.



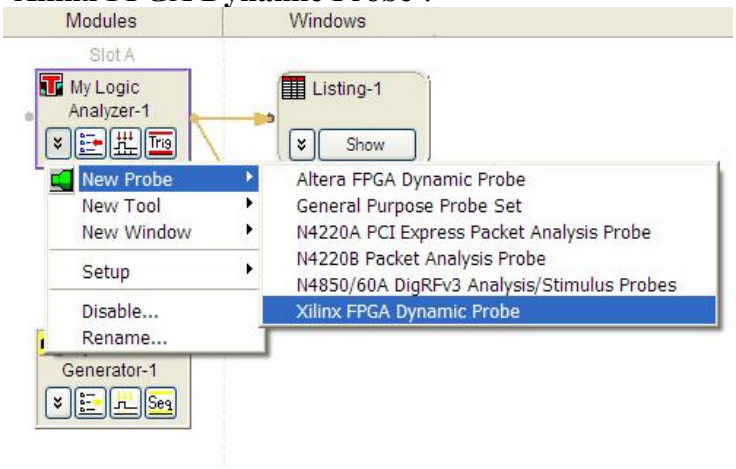


Xilinx Spartan-3E board pin location	LA Pod Channel
IO1	Ch7
IO2	Ch0
IO3	Ch1
IO4	Ch2
IO5	Ch3
IO6	Ch4
IO7	Ch5
IO8	Ch6
IO9	Clk
IO10	Ch8
IO11	Ch9
IO12	Ch10
GND	Gnd

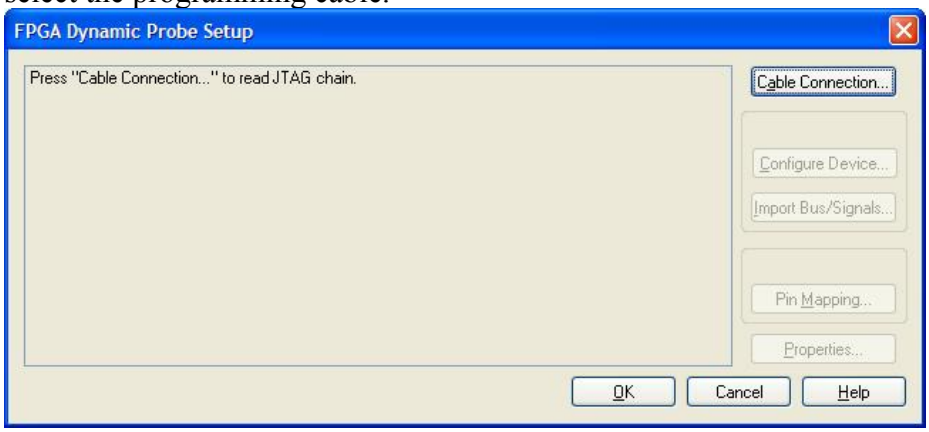
Remarks:

a) **The demo contains of 4 Banks:**
   i. Bank0 – 8-bit down counter + 3 signals from Button IO (BTN North, BTN East, BTN South)
   ii. Bank1 – 8-bit up counter + 3 signals from Switch IO (SW0, SW1, SW2)
   iii. Bank2 – Sine generation
   iv. Bank3 – Picoblaze address, 8KHz update

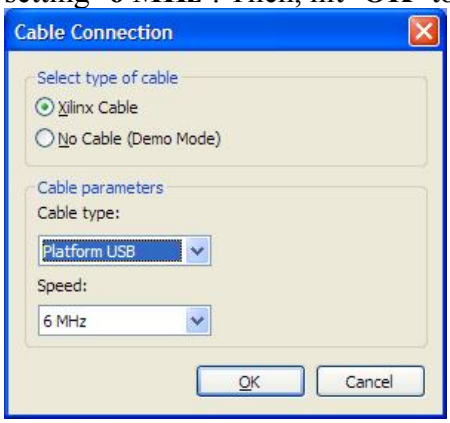
6.2 To add the Xilinx FPGA Dynamic Probe the user will right click on the Logic Analyzer in the Overview tab. From the pop up select '**New Probe**' then '**Xilinx FPGA Dynamic Probe**'.



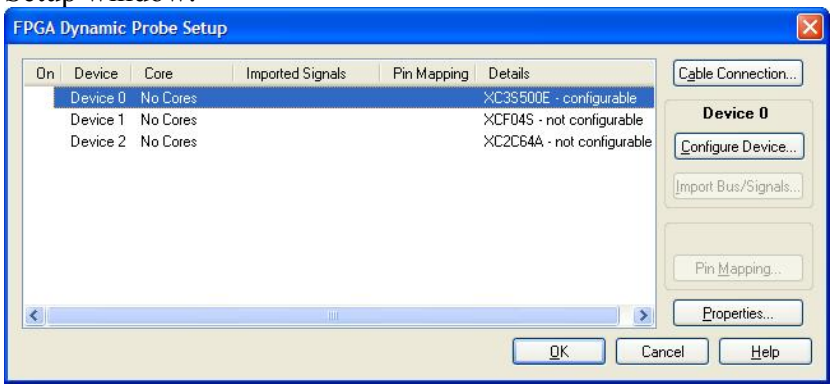
6.3 The FPGA Dynamic Probe Setup window will automatically appear on the screen as per the figure shown below. Click on the '**Cable Connection**' to select the programming cable.



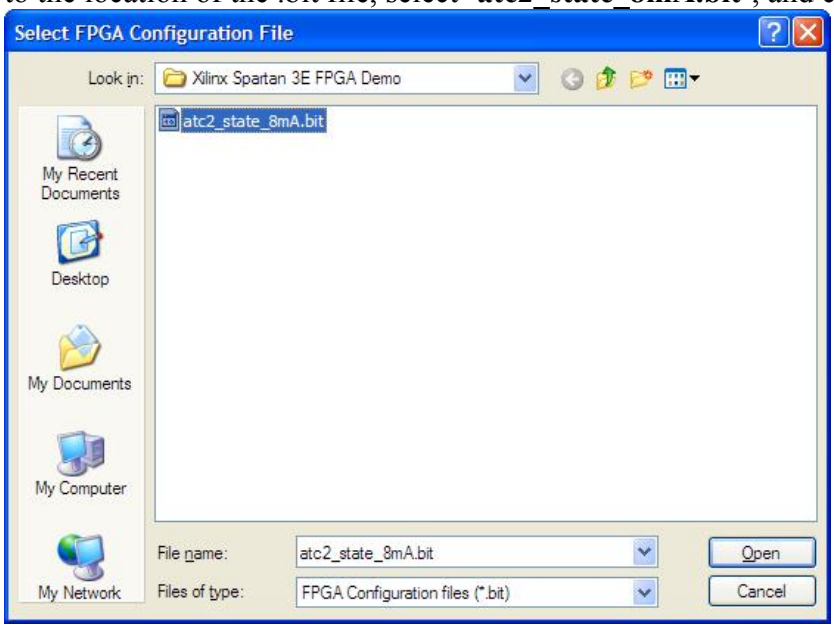
6.4 Select '**Platform USB**' from the drop-down list and leave the speed as default setting '**6 MHz**'. Then, hit '**OK**' to continue.



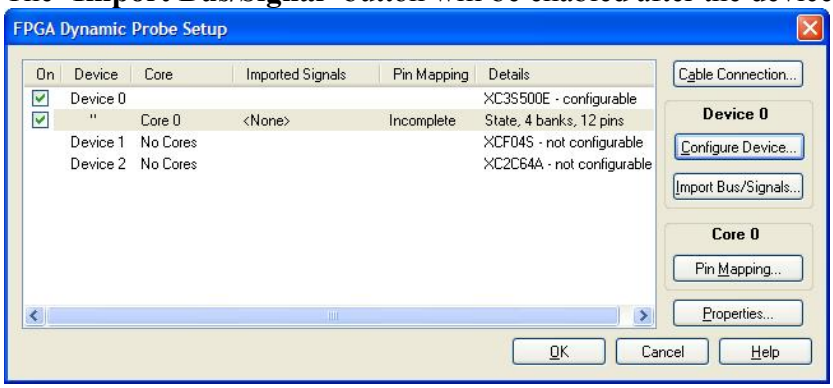
6.5 Devices on the JTAG chain will be displayed on the FPGA Dynamic Probe Setup window.



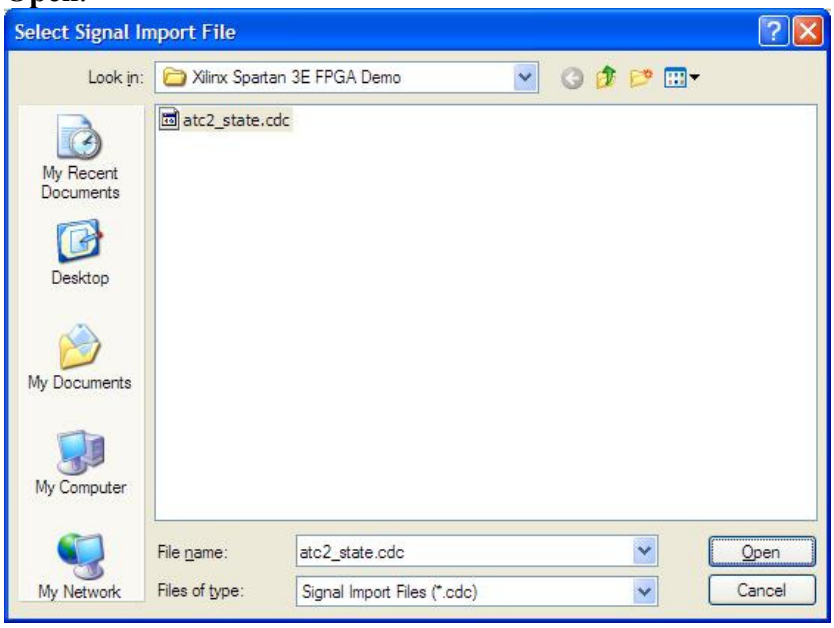
6.6 Click on '**Configure Device**' to select the FGPA configuration file. Navigate to the location of the .bit file, select '**atc2_state_8mA.bit**', and click '**Open**'.



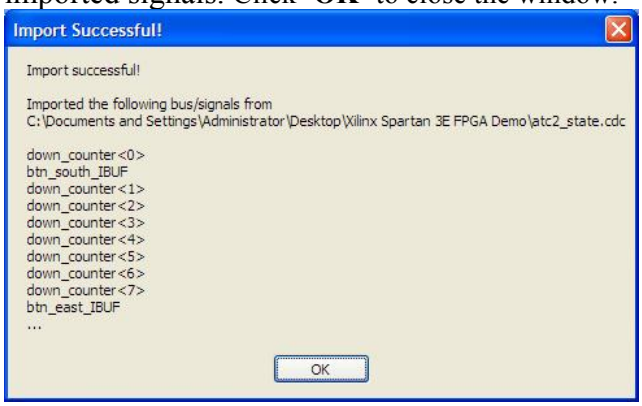
6.7 The '**Import Bus/Signal**' button will be enabled after the device is configured.



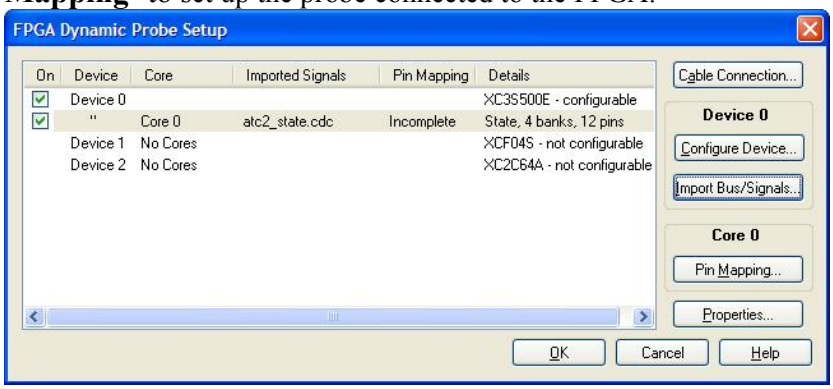
6.8 Click on '**Import Bus/Signals**' to select the Logic Analyzer Interface file. Navigate to the location of the .cdc file, select '**atc2_state.cdc**', and click **Open**.



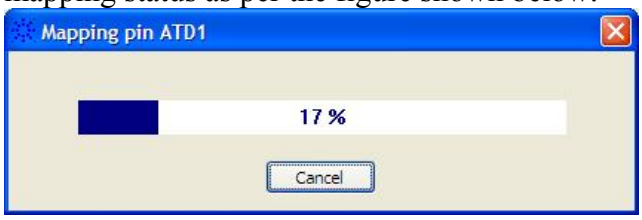
6.9 A pop-up window will show the import status and display a subset of the imported signals. Click '**OK**' to close the window.



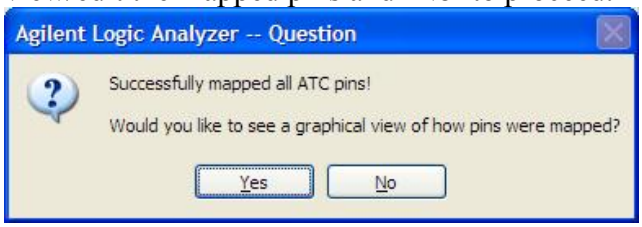
6.10 The setup window will display the core details: number of pins, number of banks, and capture mode. Select the required core and click on the '**Pin Mapping**' to set up the probe connected to the FPGA.



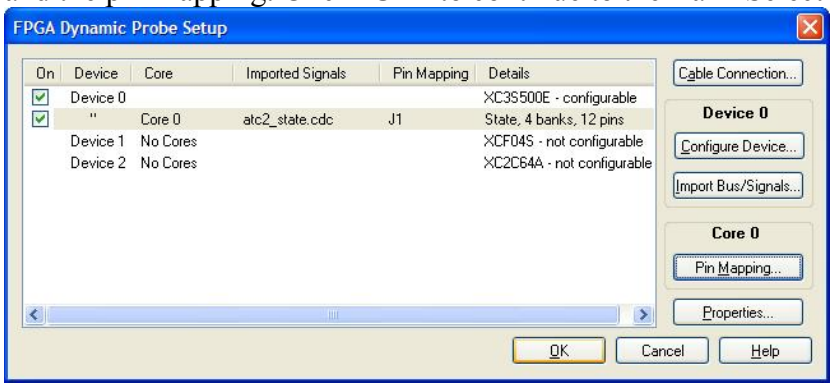
6.11 The system will proceed with the automatic pin mapping and showing the pin mapping status as per the figure shown below.



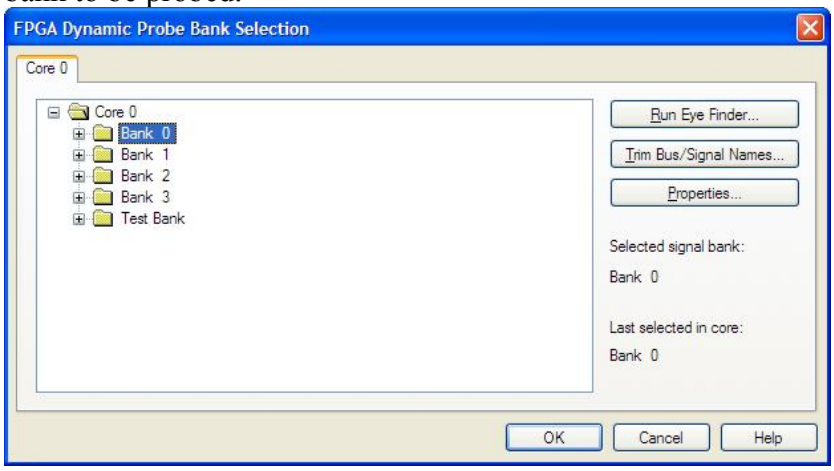
6.12 When the pins are successfully mapped, a message will prompt. Click '**Yes**' to view/edit the mapped pins and '**No**' to proceed.



6.13 The Setup window should now display the core details, the imported signals, and the pin mapping. Click '**OK**' to continue to the Bank Selection.



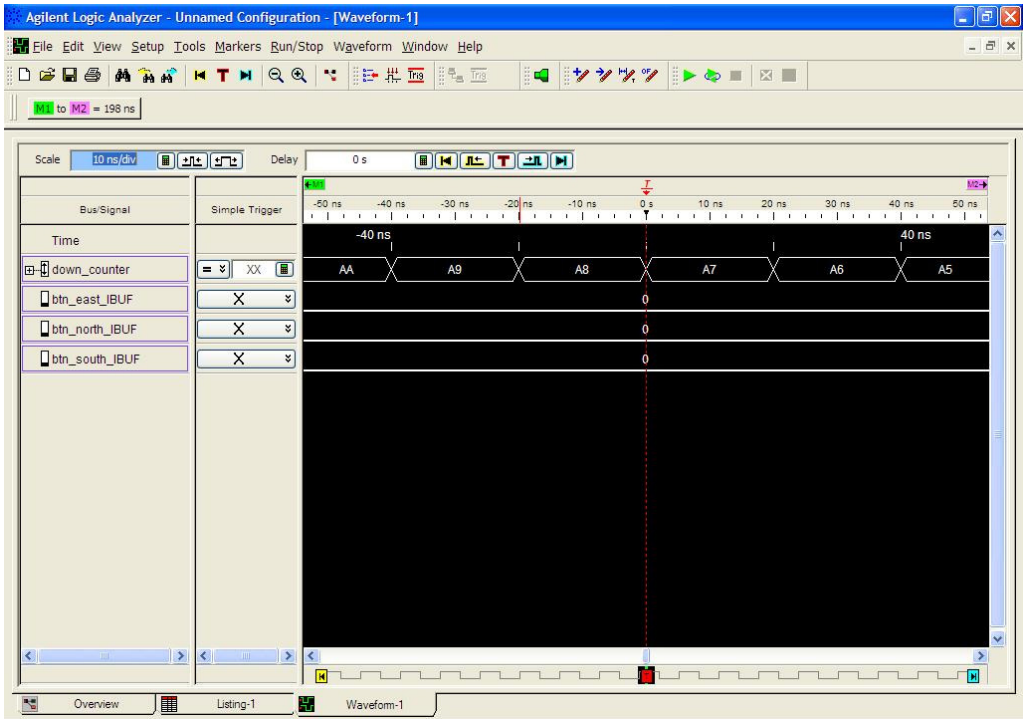
6.14 In the FPGA Dynamic Probe Bank Selection window, the user can select the bank to be probed.



6.15 The system is now set up to debug the FPGA design.

Note:

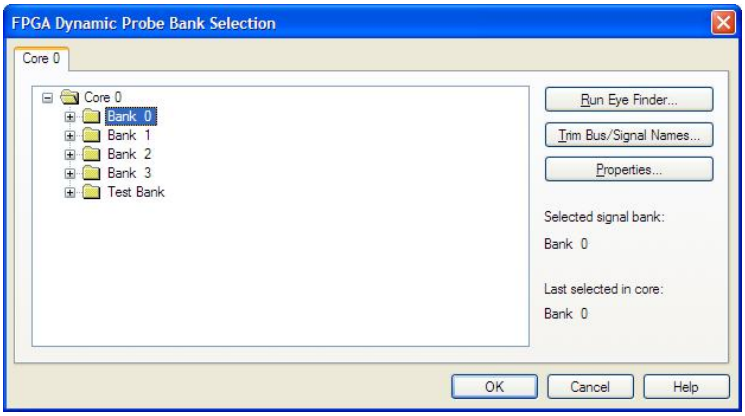
1. **The Logic Analyzer will be setup according to parameters read from the LAI file. This included clocking, labels, and capture mode. The user can now modify acquisition depth and set up the Logic Analyzer trigger.**
2. **You can toggle the 'Button IO' (BTN North, BTN East, BTN South) signals in Bank 0 and 'Switch IO' (SW0, SW1, SW2) signals in Bank 1. This demonstration proves the LA is capturing the live signal running in the FPGA.**
3. **Set to 'View As Chart' for Bank 2. You will actually see a sine waveform on the 'Waveform' window.**



6.16 To switch your bank, you just need 4 clicks. Follow the following steps.

Note:

1. **Can you imagine Tek's solution needs 22clicks and requires 3 separate applications.**

1st Click	Click on the  '**Probe Properties**' icon at the toolbar.
2nd Click & 3rd Click	Select your bank and click on the '**OK**' button to finish the bank selection. 
4th Click	Click on the  '**Run**' button at the toolbar to acquire the data.