Version for EDK 10.1.03 as of June 14, 2010

Goals

- To understand how to utilize chipscope in debugging the FPGA
- To learn how to incorporate different debugging chipscope cores in the system
- To be able to interpret the signal waveform outputs in debugging

Requirements

You'll need access to:

- The Xilinx EDK 10.1 and ChipScope Pro 10.1 software with IP Update 3.

Preparation

Please follow the menu link below to access to the documents in XPS:-

Help⟶EDK Online Documentation⟶IP Reference⟶Processor IP Catalog

Read the following chipscope peripherals to get familiar with chipscope:-

- ChipScope ICON
- ChipScope ILA

Note

Using ChipScope in EDK does not require us to generate the core manually. The cores are integrated into the tool using TCL script. The TCL script is called during the execution of EDK Platgen tool. The script calls Core Generator in command line mode and provides arguments file to generate netlist internally.

Configure Hardware

1. Using the **Base System Builder Wizard (BSB)**, create a new XUPV2P microblaze design with frequency 100.00Hz.
2. Select **On-chip H/W debug module** with local memory 32 KB and disable cache setup.
3. In the **Configure IO Interface**, select the following peripheral:
   - RS232_Uart
   - LEDs_4Bit
   - PushButton_5Bit
4. In the **Software setup**, uncheck both *Memory test* and *Peripheral Selftest*.
5. Continue the BSB setup and click **finish**.
6. Browse to **IP Catalog⟶Debug**, double click on *ChipScope Integrated Logic Analyzer (ILA)* and *Chipscope Integrated Controller (ICON)* to add them to the system.

7. Scroll to **System Assembly View⟶Ports**, make new connections for the following ports:-
   - *LEDs_4Bit⟶GPIO_IO_O*
   - *PushButton_5Bit⟶GPIO_IO_I*
   - *chipscope_icon_0⟶control0*
8. Under the *chipscope_ila_0* tab, connect the following ports:-
   - *TRIG0* connects to *PushButtons_5Bit_GPIO_IO_I*
   - *DATA* connects to *LEDs_4Bit_GPIO_IO_O*
   - *CLK* connects to *sys_clk_s*
   - *CHIPSCOPE_ILA_CONTROL* connects to *chipscope_icon_0_control0*
9. Double click on *chipscope_ila_0* and make the following changes:-

   Under the Misc tab
   - Change *Number of Data Samples Captured Per Trigger Match* to 4096.
   - Disable *Use The ILA Trigger Signal As The ILA Data Signal* to allow signal other than the trigger as data output.
   - Change *Width of ILA Data Input Signal* to 4 as the LEDs_4Bit will be the outputs
   - Uncheck *Enable The Trigger Out Signal "trig_out"*
   - Change *Maximum Number of Levels supported by the Trigger Sequencer* to 4

   Under the Trigger Unit tab
   - Change *Number of Match Unit Enabled for Trigger Unit 0* to 4
   - Change *Width of Trigger Input for Trigger Unit 0* to 5
   - Leave the *Match Counter Width* and *Match Type* as default. (Note: You may refer to the documents in **Preparation** section to learn more comparison operations that are provided with different match types. For now, we use only == and $\prec\succ$ operations in basic mode.)
10. Generate bitstream and make sure there is no errors.

Configure Software

1. Scroll to **Software** tab and click **Add Software Application Project**, name it as **chipscope**.
2. Under **Applications** tab, add the *chipscope.c file* as your source file in the project directory .
3. Under **Software** tab, click **Build all user application** and make sure there is no errors.
4. Read the simple *chipscope.c* file and predict the shape of the waveforms.
5. Download the bistream to your board and launch XMD shell in Hardware debugging mode.
6. After the connection is established, download the source file with **dow chipscope/executable.elf** and type **run**.

Using Chipscope Analyzer

1. Launch the Chipscope **Analyzer** under your ChipScope Pro directory.
2. Click on the *Open Cable/Search JTAG Chain* icon at the left upper side of the window.
3. After the socket connection is opened, you should be able to see a window listing the connected devices. Make sure that XC2VP30 is connected and click **OK**.
4. Scroll to **File⟶Import**, click **Select New File**, browse to your project directory and select chipscope_ila_0.cdc file in your root folder. (Note: If you cannot find the file, browse to ....(*project directory*)/implementation/chipscope_ila_0_wrapper/chipscope_ila_0.cdc instead.)
5. In the **Trigger Setup⟶Match Functions** interface, change **M0** value from X_XXXX to 0_XXXX. This means that the **ENTER** push button will treat as the trigger. (Note: Look at the chipscope.c file to identify which bit correspond to which input signal.)
6. Click the "Play" icon on the shortcut command bar. No waveform will be displayed until the match unit is trigger.
7. Now, press the **ENTER** push button as trigger. When the signal change to low, 4096 cycles of data (as specified in EDK) are collected and the wavaform is displayed. Since the default constant is 0, all the signals should be low.
8. To display a +1 up counter, click "Play" icon again, press and hold the **UP** push button. Before you release, press **ENTER** to trigger the output collection. You should be able to see the up counter waveform right now. (Note: Do the same for different push buttons to change the counter constant and observe different waveform.)
9. The LEDs light up all the time as the signal changes are too fast to be observed.
10. You can use the sequencer to trigger the chipscope in any orders you like. For instance, change the following match units:-

Match Unit	Value	
M0	X_0XXX	Up
M1	X_X0XX	Down
M2	X_XX0X	Left
M3	X_XXX0	Right

11. In the **Trigger Setup⟶Trigger Conditions** interface, double click on **MO** under **Trigger Condition Equation**, enable all four match units in **Boolean** tab, switch to **Sequencer** tab and assign the following match units:-

Level	Match Unit
1	M0
2	M1
3	M2
4	M3

12. To trigger the chipscope **M0⟶M1⟶M2⟶M3**, press the buttons in this order **UP ⟶DOWN ⟶LEFT ⟶RIGHT**.
13. As long as these conditions happen in sequence, chipscope can be triggered. It means even **UP ⟶ ENTER ⟶DOWN ⟶ENTER ⟶LEFT ⟶ENTER ⟶RIGHT** can be the trigger because the four conditions happen.
14. Try on other match units and sequencer to get familiar with chipscope.

Reference

You may refer to the following materials for further information about chipscope:-

- ChipScope Pro 10.1 Serial I/O Toolkit User Guide
- ChipScope Pro 10.1 Software and Cores User Guide