Using ChipScope with Xilinx Platform Studio (XPS)

Kazi Asifuzzaman

asifuzzaman.kazi.186 @ student.lu.se

Lund University, Sweden – August 19, 2010

Introduction :

A step by step procedure to use ChipScope with a basic system developed in Xilinx Platform Studio (XPS).

Requirements:

- i. Xilinx ISE Design Suite 12.1
 - A. Xilinx Platform Studio (XPS)
 - B. ChipScope Pro
- ii. Digilent Adept 2.4 or Higher (Windows)
- iii. Digilent Nexys 2 Board
- iv. Digilent Plug-in for ISE 12.x
- v. Download Cable

Work Flow:

- 1. Xilinx Platform Studio (XPS)
 - A. Creating a system from the scratch using Base System Builder (BSB)
 - B. Getting used to with different parts of XPS
 - C. Adding debugging cores into the existing design
- 2. Digilent
 - A. Installing necessary software and plug-in for the target board to be properly connected
 - B. Using Digilent Adept for downloading bitstream to the board.
- 3. Using HyperTerminal to 'talk' with the board.
- 4. Chip Scope Pro
 - A. Synchronize an existing design to a new ChipScope project.
 - B. Observing bus transactions.

1. Xilinx Platform Studio (XPS):

1.1 Creating a new design with BSB:

To start, open Start > All Programs > Xilinx ISE Design Suite 12.1 > Xilinx Platform Studio>

i. As soon as XPS starts 'Create New or Open Existing Project' wizard shows up.

🗢 Xilinx Platform Studio 🛛 🛛 🗙
Create new or open existing project
858 ③ Base System Builder wizard (recommended)
Blan(Create a Base System using the BSB Wizard. This is the
Open a recent project
Browse for More Projects
Browse EDK examples (projects) on the web here

ii. Select the 'Base System Builder' (BSB) option. This would allow you to build a basic working design only with few clicks. However, you have to check/change default settings according to your design needs. Click 'OK'.

🗢 Create New XPS Project Using BSB Wizard	Browse For Folder	
New project Project file C:/Kazi/TestProjects/A02ChipScopeDual/system.xmp Browse Advanced options - optional (press F1 key for help) Set Project Peripheral Repository Search Path Browse OK Cancel	Add Repository Directory to List of Directories	×

iii. In this window you specify your 'project folder path' and 'project peripheral repository search path'. It is always wise to make a new folder with a relevant name that best describes the project and place the system.xmp file in that folder. It is also recommended that you make the project folder on the hard drive of the computer rather than on a USB memory or any other removable storage device. In this window also locate the support files for the target board as shown. Click 'OK'.

iv. The Base System Builder (BSB) wizard starts. It will take you through rest of the steps of creating a new embedded system design. At the first step select 'I would like to create a new design'. Click 'Next'.

Board		
 I would like to cr 	eate a system for the following development board	
Board Vendor	Digilent	~
Board Name	Nexys 2-1200 Board	*
Board Revision	C	~
🔘 I would like to cr	eate a system for a custom board	

- v. Select the board information from the drop down menus as shown. Click 'Next'.
- vi. On the system configuration step select 'Single Processor System'. Click 'Next'.
- vii. Configure processor 1 with setting shown. Click 'Next'.

-Processor 1 Configuration		
Processor Type	MicroBlaze	~
System Clock Frequency	50.00	MHz
Local Memory	32 KB	~
Debug Interface	On-Chip HW Debug Module	~
Enable Floating Point	Unit	

- viii. Keep default values for 'Peripheral' and 'Cache' steps.
- ix. In the 'Application' step, under 'Peripheral Test' submenu change the operation values for 'Instructions' and 'Data' to 'ilmb_cntlr' and 'dlmb_cntlr' respectively.

Welcome	Board	System	Processor	Peripheral	Cache	Application	Summa
pplication Configura	tion						
onfigure the example ap	plications.						
Example Applications		Contras United					
Application		Option Value					
Test microblaze_0							
- Standard IO		RS232_PORT					~
- Boot Memory		ilmb_cntlr					~
🖨 Memory Test		TestApp_Mem	ory_microblaze_0				
Instruction:	5	ilmb_cntlr					~
Data		dimb_cntir					~
🛓 Peripheral Test		TestApp_Peri	oheral_microblaze_0				
- Instruction	5	ilmb_cntlr					~
Data		dlmb_cntlr					~
Interrupt W	actor	ilmb_cntlr					

- x. Have a look at the System Summary. Notice the components listed their addresses and location of project files. Click 'Finish'.
- xi. Your initial system has been created. Select 'Start using Platform Studio' in the next wizard.

1.2 Exploring Xilinx Platform Studio:

It is good to know few things before actually start working with XPS.

i. Block Diagram:

A block diagram of your new design should already been generated otherwise you can generate it from **Project > Generate Block Diagram Imag**e.



A Block diagram shows an overview of the components and how they are connected in the system.



Spend few minutes on the block diagram and notice the different parts of the design like BRAM, microblaze_0, FSL, PLB, LMB, GPIO, mdm_0 etc.

- ii. Left Panel Project, Applications & IP Catalog:
 - Project Tab: The project references to project related files. Information is grouped in three categories:
 - a. <u>Project Files:</u> Project specific files such as the Microprocessor Hardware Specification (MHS) files, Microprocessor Software Specification (MSS) files, User constraint file (UCF) files etc. Have a look at each of the files to have some general idea on these.
 - b. <u>Project Options:</u> Project specific options such as device, Netlist, Implementation, Hardware Description Language (HDL) and Sim Model Options.
 - c. <u>Design Summary</u>: A graphical display of the state of your embedded design and gives you easy access to system files.
 - Applications Tab: This tab lists the software application option settings, header files, and source files that are associated with each application project. With this tab selected you can:
 - a. Create and add a software application project, build the project, and load it into the block RAM.
 - b. Set compiler options
 - c. Add source and header files to the project.
 - IP Catalog tab: This tab lists all available IP cores (including their licensing status, release version, supported processors and classification)those can be added to a embedded design.
- iii. Right Panel System Assembly View:

System Assembly View allows you to view and configure system block elements. It has three tabs:

- Bus Interface: Displays the buses in your design. Use this view to modify the information and connections for each bus.
- Ports: Displays ports in your design. Use this view to modify the details for each port.
- Addresses: Displays the address range for each IP instance in your design . Click 'Generate Address' to automatically generate the system address map.
- iv. Activate the other Software Project (which activates and uses LEDs, Switches, 7 Segment LED etc. of the FPGA board) by Initializing BRAM for it.



v. To automatically configure and generate library code for your system click **Software > Generate Libraries and BSPs.**



This step also generates header files containing symbolic names for memory mapped peripherals. Run this step and look at your software projects (Applications tab in the left frame). Click on the + left of *processor:microblaze_0* in your active project. There you will find a header file (xparameters.h) with symbolic names of all memory mapped peripherals.

- vi. The standard C function **printf** generate huge libraries which will not fit into the small onchip memory. Instead use **print** or **xil_printf** function to output text. You can compile your programs by selecting **Software > Build All user Applications.**
- vii. Click Hardware > Generate Bitstream. It might take long to finish (even more than 10 minutes).



Fortunately the hardware is independent of the software, so whenever you change your C files, this step will not rerun. During the hardware synthesis, the tools use a User Constraint File (UCF), which can be found in the left frame in the Project tab. This file binds external ports in your design to physical pins on the FPGA. It is important that the names in the design are the same as in this file, so do not rename the external ports. Also if you add or remove external pins this file must be edited, i.e. removing a GPIO peripheral.

viii. The last step before downloading the configuration to the FPGA is to merge software binaries and the bit stream from the hardware synthesis. Run **Device Configuration > Update Bitstream.** This places the executable in the on chip memory making the configuration file ready to be downloaded in the FPGA.

1.3 Adding Debugging Cores into Design:

- i. ChipScope Integrated Controller (ICON): It provides communication with other ChipScope cores. You must use ChipScope ICON core to use any of the other ChipScope cores, because it provides the JTAG connectivity for all other ChipScope cores. One ICON can connect to one or more ChipScope cores via one or more 36-bit control connections. The On-chip Peripheral Bus (OPB) or Processor Local Bus (PLB) Integrated Bus Analyzer (IBA) must be connected to the bus using a monitor (Bus Analyzer) connection.
 - a. To add a ICON core to the design, open the '**IP Catalog**' tab. Unfold the '**Debug**' menu and locate '**ChipScope Integrated Controller**'. Select and right click on the IP, click '**Add IP**'.

atalog 🔹	+□₽×	Bus Interfaces	Ports	Bus Interfaces	Ports	Addresses		
•		Name		Name			Net	
cription		External Ports		😟 External Ports				
EDK Install		± dlmb		🛨 dimb				
Analog		🕀 ilmb		🛨 ilmb				
Bus and Bridge		. mb_plb		⊕ mb_plb				
Clock, Reset and Interrupt		∃ microblaze_0		microblaze_0				
Communication High-Speed		⊞ Imb_bram		🛨 Imb_bram				
Communication Low-Speed		🗄 dimb_cntir		😟 dlmb_cntlr				
DMA and Timer		🗄 ilmb_cntir		🗄 ilmb_cntir				
- Debug		Micron_RAM		Micron_RAM				
Agilent Trace Core		🛨 mdm_0		庄 mdm_0				
		🗄 LED_75EGMENT	r	∃ LED_7SEGMENT				
Agilent MicroBlaze v5 Trace Core		🕀 LEDs_88it		🛨 LEDs_88it				
		🗄 Push_Buttons_1	3Bit	Push_Buttons_3.	Bit			
Chipscope Integrated Controller	Add IP		CONTRACTOR DE LA CONTRACTÓR	🛨 Switches_BBit				
🚽 📩 📩 Chipscope Integrated Logic Analyz								
🚽 🚽 📩 Chipscope PLBv46 Integrated Bus	View MPD			😑 chipscope_icon_	0	In the second second second		1
— 🚖 Chipscope Virtual IO (VIO)	View IP M	odifications (Change L	.00)	- control0		Configure IF	Second contraction and the	l
🚽 🚽 📩 🚽 MicroBlaze Debug Module (MDM)		Datasheet		- tdi_in				
🔄 👉 🏰 Xilinx MicroBlaze Trace Core (XMTC	VIEW PDF	Datasrieet		- reset_in		View MPD		
General Purpose IO	Make This	TR Local				View IP Mod	ifications (Change Log)	
IO Modules	Pidito Thi	i cocu		- update_in		View PDF Da	atasheet	
Interprocessor Communication				- sel_in				-
Memory and Memory Controller				- drck_in - tdo out		Delete Insta	ince	
DE PCI				too_out	0			-
🖶 Peripheral Controller				± clock_generator ± proc_sys_reset		Make This IF	P Local	
+ Processor				E proc_sys_reset_		The second		

- b. Notice that, in the 'System Assembly Panel', under Ports tab a new peripheral is added named 'chipscope_icon_0'. Right click on the instance and click 'Configure IP'.
- c. In the appearing configuration wizard set the value of '**Number of Control Ports**' as 2. This means this ICON can now be connected to two other ChipScope cores.
- d. You can check the **system.mhs** file at **Project** tab has now been updated with following modifications:

```
BEGIN chipscope_icon
PARAMETER INSTANCE = chipscope_icon_0
PARAMETER HW_VER = 1.04.a
PARAMETER C_NUM_CONTROL_PORTS = 2
END
```

ii. Integrated Logic Analyzer (ILA): ILA is used to monitor individual signals in a processor design. When the ChipScope ILA core is used in Platform Studio, only signals at the top level of the processor design (at the Microprocessor Hardware Specification (MHS) level) can be monitored using the ILA. Using the ILA connection feature in the FPGA Editor, you can monitor signals at level of hierarchy using this ILA.



a. To add a ILA core to the design click **Debug > Debug Configuration.**

Add New ChipScope Peripheral	X
Optionary area on by added to monitor the various algound made the by adding a series core: In monitor R3 +4.5 but algound (adding R3 BA) In monitor R3 +4.5 but algound (adding R3 BA) In monitor R3 +4.5 but algound (adding R3 BA) In monitor R3 +4.5 but algound (adding R3 BA) In monitor R3 +4.5 but algound (adding R3 BA) In monitor R3 +4.5 but algound (adding R3 BA) In monitor algoung synthemical algound (adding R4) In the provide algoung a	All Alting run-films. Filease induces what you went to anthere and the properties a registry of the properties a registry of the properties a registry of the properties and the proper
	Or Carol

- b. Notice the diagram on the appeared window. It may help you to understand the connection and organization between ChipScope cores. Click 'Add ChipScope Peripherals'.
- c. Select 'To monitor arbitrary system level signals (adding ILA)' from the list.
- d. Configure ILA and select which signals you wish to monitor with it.

System	Basic Advanced	
Original Hardware Synde Original (6.0) Original (6.0)	Available Ports on Instance: microblase_0 FSI0_5_CLX (Connected to microblase FSI0_5_ELX) (Connected to microblase FSI0_5_ELX) (Connected to FSI_0_M) FSI0_5_CLX (Connected to FSI_0_M) FSI0_5_CLX (Connected to FSI_0_M) FSI0_5_M_VMTRF (Connected to FSI_0_M) FSI0_5_M_VMTRF (Connected to FSI_0_M) FSI0_5_M_VMTRF (Connected to FSI_0_M)	d by ChipScope ILA (Integrated Logic Analyser). Signals Monitored by: TR100 V (d, 50,0000Her, 198, 2010, 0.0, 1, fopa, 0, 2432, 244, 244, 244, 244, 244, 244, 24
	P311, M_CAK (Connected to P5, M_K P311, M_CONTROK (Connected to F5 P311, M_DATA (Connected to F5, M, P311, M_VARTE (Connected to F5, M, P311, S_CAK (Connected to F5, S, C P311, S, REAP (Connected to F5, M, S) F312, M_CAK (Connected to F5, M, S)	¢X
	Select the number of signal samples you want Select the clock to be used on monitoring:	c1k_50_000013fz
	Select the clock to be used on monitoring:	e11_50_00000016 V I should be equal to or fester than the frequency that it ports can be monitored.
	Select the clock to be used on monitoring: Notes • The clock used to monitor the sign signal is operating at. • For each IP indiance, only its output • Live TRIGO finely, and then TRIGI, its	e11_50_00000016 V I should be equal to or fester than the frequency that it ports can be monitored.
Add Chysicope Perpherel Delete Chysicope Perpherel	Select the dock to be used on monitoring: Index • The dock used to monitor the signal • The dock used to monitor the signal • The dock used to monitor the signal • Use third of equilable the TMGs () • Use third of equilable the TMGs () • If you want to debug bus signals, u	e11_50_00000016 V I should be equal to or fester than the frequency that it ports can be monitored.

- iii. Integrated Bus Analyzer (IBA): Facilitates monitoring of Processor Local Bus (PLB) transactions.
 - a. To add a IBA core to the design click **Debug > Debug Configuration.**
 - b. In the window appeared Click 'Add ChipScope Peripherals'.
 - c. Select the option 'To monitor PLB v4.6 bus signals (adding PLB IBA)'.
 - d. Configure IBA as shown in the following figure:

stem	Basic	Advanced						
Monitor Hardware Signals chipscope_ila_0 chipscope_plbv46_iba_0 Debug Software Application	Chip50 bus.	cope IBA (Integrated Bus	s Analyzer) is mainly used to monito	r the transaction that happens on PLB				
- Debug Sortware Application	-Mon	itor Bus Signals						
Miscellaneous								
JING DAKT	Please select the bus that you want to monitor:							
	Plea	ase select the signals on I	the bus that you want to monitor:					
		🗹 Bus Control Signals	🗹 Bus Read Data Signals					
		Bus Address Signals	🗹 Bus Write Data Signals					
		Enable Hardware/Softwa	re Co-debug					
	Tar	get Processor:		~				
		; will set up the mutual tri bling software/hardware	iggering between the software deb : co-debug.	ugger and hardware signals,				
	Select the number of signal samples you want to collect:							
	Note Chij sam		nal samples to create waveforms th M.	nat help you debug. The collected				
	Information	n:						
Add ChipScope Peripheral	Number of	BRAMs being used: 23						

- e. Click 'OK'.
- f. You can also check and modify configurations setting for these cores directly into **system.mhs** file under project tab.
- g. Always remember to perform **Device Configuration > Update Bitstream** after doing any modification in the design to ensure that FPGA gets the updated configuration.

2. Digilent:

2.1 Initializing Cable Connection to the Board:

- i. Connect the board properly with both USB and RS232 cables. And switch the power on.
- ii. Microsoft 'Found New Hardware' starts. Select 'Yes, this time only'. Click 'Next'.
- iii. Select 'Install the Software Automatically (Recommended)'. Windows will now import some files automatically to install the new device.
- iv. Install Digilent Adept 2.4 (If not already installed)

2.2 Installing Digilent Plug-in for ISE 12.x:

- i. Copy the plug-in files "**libCseDigilent.dll**" and "**libCseDigilent.xml**" into the ISE Design Suit Installation.
- ii. For the ISE Design Suite, the typical location is

C:\Xilinx\12.1\ISE_DS\ISE\lib\nt\plugins\Digilent\libCseDigilent Note: For 64-bit Windows, use nt64 in place of nt.



- 2.3 Downloading the bitstream to FPGA using Digilent Adept :
 - i. Start 'Digilent Adept'. Select 'Onboard USB' from the drop down menu in 'Connect:' field.
 - ii. Browse the option beside its written FPGA XC3S1200E and select the **download.bit** configuration file from your project directory home under implementation/. Click '**Program'**.
 - iii. Your FPGA is now programmed with your design configuration.

🛆 Digilent Adept		×	Open			E.	? 🗙
NEXYS 2	Connect: Onboard USB		Look in:	implementation	🖌 G 👂 🖻		
	Product: Nexys2 - 1200			🚞 _xmsgs	micron_ram_wrapper		
		_	Mv Recent	Cache	proc_sys_reset_0_wrapper wrapper wrapper		
Config Test Register I/O File I/O I	O Ex Settings	-	Documents	dimb cntir wrappe			
	1			🚞 dimb_wrapper	cswitches_8bit_wrapper		
FPGA XC3S1200E	Browse Program			imb_cntir_wrappe	r 🚞 xinx_auto_0_xdb		
AC3372002			Desktop	imb_wrapper led 7segment wra			
PROM	Browse Program			leds_8bit_wrapper	r		
XCF04S				Imb_bram_wrappe Imb_bram_wrapper	er.		
			My Documents	mo_pio_wrapper			
				microblaze_0_wrap	pper		
			My Computer				
				File name: d	lownload.bit	Ope	n
				Files of type: F	'PGA Config Files (".bit; ".svf) 🛛 😽 😽	Cano	el 🛛
Initialize C	hain		My Network		Open as read-only		
							.::
Board information loaded. Found device ID: f5046093	<u>t</u>	-					
Found device ID: 21c2e093 Initialization Complete.							
Device 1: XC3S1200E							
Device 2: XCF04S		-					
t		-					

3. Communicating with the Board Through HyperTerminal:

To make sure that everything went right in all previous steps, we would try to communicate with the board through HyperTerminal. At this stage the board should be programmed with your design configuration (**download.bit** file has been downloaded in the FPGA board).

• Start HyperTerminal from Start Menu.

 Microsoft Update Set Program Access and Defaults Windows Catalog 				
m Accessories	•	🛅 Accessibility	×	
🛅 Digilent	≁	m Communications	•	🎭 HyperTerminal
🛅 Games	≁	🛅 Entertainment	•	📚 Network Connections
🛅 Keyspan USB Serial Adapter	≁	🛅 System Tools	•	💁 Network Setup Wizard
🛅 Microsoft Silverlight	→	💟 Address Book		🔄 New Connection Wizard
🛅 Mozilla Firefox	→	📓 Calculator		🗳 Wireless Network Setup Wizard
🛅 Norman Security Suite	•	🔤 Command Prompt		💼 HyperTerminal 🔹 🕨
🤤 OpenOffice.org 3.2	•	📕 Notepad		
🛅 Startup	≁	🦉 Paint		

• Give any name to the new connection.

Hew Connection - HyperTermin Pin Edit View Cal Transfer 1980 D @ @ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Connection Description Connection Description Connection Enter a new and choose an icon for the connection: Name Ent Con Con Con Con Con Con Con C	
Deconnected Auto detect	DK Cancel	3

- Select appropriate port of communication (COM1/COM3 etc.)
- Define port setting as below. Click OK.

COM3 Properties
Port Settings
Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None
Restore Defaults
OK Cancel Apply

• You can check 'Send line ends with line feeds' and 'Echo typed characters locally' at ASCII setup in connection properties to view sent data in the Hyper Terminal window. (Optional)

BD Properties	? 🔀
Connect To Settings	
ASCII Setup	indows keys
ASCII Sending	
 Send line ends with line feeds Echo typed characters locally 	1+H, Space, Ctrl+H
Line delay: 0 milliseconds. Character delay: 0 milliseconds.	Terminal Setup
ASCII Receiving Ascentral Append line feeds to incoming line ends	*
 Force incoming data to 7-bit ASCII Wrap lines that exceed terminal width 	or disconnecting
OK Cancel	ASCII Setup
	OK Cancel

• Now, press the BTN0 button on the FPGA board. HyperTerminal window should show following output:

🎨 BD - HyperTerminal	
File Edit View Call Transfer Help	
Entering main() Starting MemoryTest for Wicron_RAM: Running 32-bit testPASSED! Running 8-bit testPASSED! Exiting main()	
Connected 0:00:14 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo	

• If you don't get this output on HyperTerminal window check back or redo your previous steps.

4. Chip Scope Pro:

Once you are done with all the previous steps of this manual then you can proceed using the ChipScope Pro analyzer. At this point it is assumed that you have the target board connected with the system and it has been programmed with your configuration (**download.bit** file has been downloaded in the FPGA board).

• Start ChipScope Pro Analyzer from Start Menu.



• To connect the board select JTAG Chain > Open Plug-in.

ChipSco	pe Pro	Analy	zer [nev	w project					
<u>F</u> ile ⊻iew	<u>J</u> TAG	Chain	<u>D</u> evice	<u>₩</u> indow	<u>H</u> elp				
# 🕑		Server	<u>H</u> ost Set	tting					
New Project									
JTAG Chain	 Xilinx <u>P</u>arallel Cable 								
	0	 Xilinx Platform USB Cable 							
	۲	<u>O</u> pen l	Plug-in						
		<u>C</u> lose	Cable						
		Get Ca	able <u>I</u> nfori	mation					
	Ø	<u>A</u> uto C	ore Statu	is Poll					

• In the appeared window write down 'digilent_plugin' in the text box for Plug-in Parameters. Click 'OK'.

ChipSco	pe Pro Analyzer [new project] Open Plug-in	\mathbf{X}
?	Plug-in Parameters-	
	OK Avbryt	

• Check if the JTAG devices appeared correct in the new window. Click 'OK'.

JTAG C	Chain Device Order				
Index	Name	Device Name	IR Length	Device IDCODE	USERCODE
0	MyDevice0	XC3S1200E	6	21c2e093	
1	MyDevice1	XCF04S	8	f5046093	

• Import .cdc file for IBA core by **File > Import** and locating corresponding address of the file under implementation/ of the project directory; as shown in the following figure. Click OK.

🗐 Signal Impor	t	×
_Import File		
File:	chipscope_plbv46_iba_0.cdc	
Directory:	$\label{eq:c:Kazi} C: \cite{Kazi} Test Projects \cite{A02ChipScopeDual} with the product of the test of t$	
	Select New File	
Unit/Device		
	DEV: 0 UNIT: 0 (ILA)	
	Auto-create Buses	
	OK Cancel	

 You can now see array of buses from your design in the waveform window. Group the array of same bus as one single bus (select all > right click > move to bus > new bus). Do the same for every multi-bit signals.

🕲 Waveforr	n - DEV:0) MyDev	/ice0 (X	C3	S1200E) UNIT:0 My	/ILAO (ILA)
Bus/Sig	nal	х	0			
- PLB_SrdE	Term	0	0			
- PLB_rdBu	irst	0	0			
- PLB_ABus	[31]	0	0			
- PLB_ABus	[30]	0	0			
- PLB_ABus	[29]	0	0			
- PLB_ABus			0		1	
- PLB_ABus	Renar	ne				
- PLB_ABus	Color	- Due - N			New Due	1
PLB_ABus	Move t Copy t			Þ	New Bus PLB ABus	
PLB ABus			Order		PLB_SrdDBus	
PLB ABus			s Ordei	,	PLB_wrDBus	
PLB ABus	Bus R	adix		•		1
PLB ABus	Auto-c	reate B	uses			
PLB ABus	Cut					
PLB ABus	Сору					
PLB_ABus	Paste	-				
- PLB ABus	Remo		n Viewe	r		
PLB ABus	oroarr		0			

• After grouping all the signals press the **T!** (Trigger Immediately) button to see the data transaction in the buses.

🕲 Waveform - DEV:0 MyDevice0 (XC3S1200E) UNIT:0 MyILA0 (ILA)								
Bus/Signal	x	0	314	315	316 	317 	318 	
• PLB_ABus_1	00000060	00000060	340	0000066	0000	1340	0000008	
⊶ PLB_wrDBus_1	00000000	00000000						
• PLB_SrdDBus_1	00000000	00000000						
- PLB_Rst	0	0						
-Bus_Error_Det	0	0						
- PLB_lockErr	0	0						

Now we will move on to trace data on a specific location using ChipScope. To do that we have to
manually configure a trigger with our desired function. Let's say we want to monitor
transactions regarding 8 Bit Switches. Then we have figure out the memory addresses being
used for this core from the 'Addresses' tab of the 'System Assembly View' of XPS.

Bus Interfaces Ports	Addresses		
Instance	Base Name	Base Address	High Address
🚊 microblaze_0's Address Map			
dlmb_cntlr	C_BASEADDR	0x00000000	0x00007FFF
ilmb_cntlr	C_BASEADDR	0x00000000	0x00007FFF
Micron_RAM	C_MEM0_BASEA	0x80000000	0x80FFFFFF
Switches_8Bit	C_BASEADDR	0x81400000	0x8140FFFF
Push_Buttons_3Bit	C_BASEADDR	0x81420000	0x8142FFFF
LEDs_8Bit	C_BASEADDR	0x81440000	0x8144FFFF
LED_7SEGMENT	C_BASEADDR	0x81460000	0x8146FFFF
xps_timer_0	C_BASEADDR	0x83C00000	0x83C0FFFF
RS232_PORT	C_BASEADDR	0x84000000	0x8400FFFF
i mdm_0	C_BASEADDR	0x84400000	0x8440FFFF

• Notice that **PLB_ABus_1** handles the transactions of addresses being used for the cores. So to observe bus transactions regarding **8 Bit Switches** we have to modify the corresponding match unit value (**M2:TRIG2** in this case; checks PLB_ABus_1) in the following way:

👹 Trigger Setup - DEV:0 MyDevice) (XC3S1200E) UNIT:) MyILAO (ILA)				
Match Unit		Function		Value		Radix
🖹 👇 🗂 MO:TRIGO		==			X000X	Bin
M1:TRIG1		==		2000(2000(2000)		
← 🚍 M2:TRIG2	IG2		TRIG2 == 8140_x000		8140_XXXX	Hex
🔶 🚍 M3:TRIG3		==		X000_X00X Hex		
- 🗂 M4:TRIG4		==			X000X_X000X	Hex
Add Act	ve		Trigger Condi	ion Name		Trigger Condition
Del C)		TriggerCor			M2
S Type: Window 🔻	Windows:		1	Depth: 1024	•	Position:
Type: Window	**indows.			Doptil. 1024	•	
o Storage Qualification:			All Data			
Sample Buffer is full						
🛞 Waveform - DEV:0 MyDevice0 (X	C3S1200E) UNIT:0 My	/ILAO (ILA)				
Bus/Signal X	0	5	10 15	20 25	30 35	40 45
← PLB_ABus_1 813FFF	F8 00002E4C	81400004)))))))))))	XXXXXXXX	XX000XXX
← PLB_wrDBus_1 00002E	40 00000370	FFFFFFF	XXXXXXXX	XXXXX	<u> </u>	
← PLB_SrdDBus_1 000000	00 0000000 📩			00000	000	
- <mark>iba_trig_in[0]</mark>	0 0					
- PLB_Rst	0 0					
- PLB_RNW	0 0					
-Bus Error Det	0 0					

- Click **Trigger Setup** > **Run** and push the **BTNO** button of the FPGA board once; which resets the program on the board allowing ChipScope to trigger and capture the desired transaction. After few seconds the waveform is displayed where the bus transactions of your target core is captured.
- Now, you can impose other conditions to examine an exact scenario. For example you want to see bus transactions for the same core but for specific dataset on PLB_wrDBus_1. First, you have to figure out which match unit value you need to change for PLB_wrDBus_1 (In this case it is M3:TRIG3). Let's say we are looking for a bus transaction on 8 Bit Switches when PLB_wrDBus_1 has a value of 0000_XXXX. Don't forget to include M3 in the Trigger Condition Equation. Click Trigger Setup > Run and push the BTNO button of the FPGA board once.

	Match	Unit	Match Unit Function Value					Radix	
	► □ MO:TRIGO			==		X000X	Bin		
7	►			==			>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	2000/2000	Bin
	∽ 🗂 M2:TRIG2			==				8140_XXXX	Hex
	🔶 🚍 M3:TRIG3			== 0000_X			0000_XXXX	Hex	
	∽ 🗂 M4:TRIG4	► 🗂 M4:TRIG4						X000X_X000X	Hex
-	Add	Active		Tric	gger Condition Nam	•			Trigger Condition
1	Del				TriggerCondition0	5			M2 && M
l									
,	Type: Window 🔻	V	Vindows:		1 Depth:	1024		-	Position:
Contino	Storage Qualification:				All Data				
_	 Sample Buffer is full 								
Ņ	Waveform - DEV:0 MyD	evice0 (XC3	S1200E) UNIT:0 MyIL						
	Bus/Signal	x	0	35	40		45	50	
		81400000	00002E4C 02E40	00000008 00002EDQ.	_X	81400000		X X000	00010/00003E4/00003
-	PLB_ABus_1								
	PLB_ABus_1 PLB_wrDBus_1	00002E4C	813FFFFC	<u> </u>		00002E4C		X X X	0000037000002E40
~			813FFFFC	<u> </u>	000	00002E4C	χ χ	<u>))</u>) <u>00000370</u> 00002E40(.
0- 0-	PLB_wrDBus_1	00002E4C		·	000	00002E4C	XX),),),	X0000037X00002E4X.
_ _	PLB_wrDBus_1 PLB_SrcDBus_1	00002E4C 00000000	0000000	·	000	00002E4C	XX	XXX	<u>\0000037</u> \00002E4\(

- You can see that an event in waveform has been captured with your desired bus transaction (if any).
- You can further enhance your trigger condition by adding more criteria. For example, if you want to see a bus transaction where the signal PLB_RNW is high in addition to previous conditions. Find out which match unit is handling the particular signal (M1 in this case, you may have to unfold the menu to see which buses are in there). Change it as you like it to see and include the condition in Trigger Condition Equation. Click Trigger Setup > Run and push the BTNO button of the FPGA board once.
- The captured waveform would show the bus transaction(s) where all conditions that you defined are satisfied.

Trigger Setup - DEV:0 MyDevice0 (XC3S120	00E) UNIT:0 MyILAO (ILA)				
Match Unit	Function	Value		Radix	
확 🗠 🗂 MO:TRIGO	==		X000X	Bin	
🗂 👇 🗂 M1:TRIG1	==		2000(_2000(_1200(_2000)	Bin	
← 🗂 M2:TRIG2	==		8140_>0000	Hex	
🗢 🚍 M3:TRIG3	==		0000_0000	Hex	
∽ 🗂 M4:TRIG4	==		X000(_X000)	Hex	
Add Active		Trigger Condition Name TriggerCondition0	Trigger Condition Equati M1 && M2 && M3		
Del		Inggerconditiono			
Type: Window Vindows	3:	1 Depth: 1024	•	Position:	
Storage Qualification:		All Data			
Sample Buffer is full					
Waveform - DEV:0 MyDevice0 (XC3S1200E)	UNIT:0 MyILAO (ILA)				
Bus/Signal X O	0 5 10) 15 20 25	30 35	40 45 50	
← PLB_ABus_1 00003EA4 00003	3E68 81400000))		<u>),),),),),),),)</u>	
- PLB wrDBus 1 00000001 00003	2E4C 00002E4C	X X	<u> </u>	<u></u>	
- PLB_SrdDBus_1 00000000 00000			00000	1000	
- iba_trig_in[0] 0	0				
PLB_Rst 0	0				
- PLB_RNW 0	1				
Due Puise Die					

- However, depending on your purpose you can also combine the conditions with an '**OR**' function instead of an '**AND**'. Where the waveform would have outputs if any of the conditions satisfy.
- You can also get finite number of samples of desired set of transactions by changing the 'Type' field in the 'Capture' tab to 'N Samples' from 'Window'. Now you will see different occurrences of your desire transaction instead of an regular waveform with all bus transactions.

Bus/Signal	х	0	50 110	5	10	15		5		10	15		5
PLB_ABus_1	00003E64	81460000	1	31460000	X		\square	Ò.	8146	0000			81460000
PLB_wrDBus_1	00000000	00000000	0	0000002		0000		000	000000			(00000004
PLB_SrdDBus_1	00000000	00000000											
iba_trig_in[0]	0	0											
PLB_Rst	0	0											
PLB_RNW	0	0											
Bus_Error_Det	0	0											
PLB_lockErr	0	0											
PLB_PAValid	0	0											
PLB_SAValid	0	0											
PLB_busLock	0	0											
PLB_abort	0	0											
PLB_Swait	0	0											
aveform captured 201				_						11			
Type: N Samples		Samples	: Per Trigg	jer:			All Da	ita		16	🗌 Tim	estamp	

• In this case the buffer may not get 'full' as you can see in the bottom of this pane. It may keep waiting for more samples even if there is none. But you can still see the samples by pushing 'Stop'.

Abbreviations:

GPIO	- General Purpose IO
LMB	- Local Memory Bus
PLB	- Processor Local Bus
RS232	- Serial Port
BRAM	- Block RAM
VGA	- Video Graphics Array
ОРВ	- On Chip Peripheral Bus
MDM	- Microblaze Debug Module
IP Core	- Intellectual Property Core
FSL	- Fast Simplex Link

References:

[1] http://www.xilinx.com/

[2] http://digilentinc.com/

[3] Xilinx Platform Studio Tutorial, Flavius Gruian & Per Anderson