ChipScope

"Looking" At Signals Inside of the FPGA

By Jason Agron

Debugging

• For SW, this is a familiar process:

- Add in "print" statements:
 - Monitor flow through code.
 - Monitor values of important variables.
- Use a debugger:
 - Monitor low-level CPU behavior.
 - Monitor memory access.
- How does one debug hardware?
 - Additionally, how does one debug SW and HW at the same time?

Monitoring Signals



- Theoretically, all signals (wires) on a traditional circuit board can be monitored.
 - Use a probe, an oscilloscope, or a logic analyzer.
- But what about the internal state of devices?
 - Registers, busses, etc.

Systems-On-Chip (SoC)



- Completely implemented in a single piece of silicon.
 - Therefore not all important signals are "exposed".
- FPGAs:
 - Same problem lots of internal logic, but not a lot of I/O ports.
- This is where ChipScope comes in handy.

ChipScope Overview

- A useful tool for monitoring arbitrary on-chip signals in real-time.
- Has two main parts:
 - Embeddable IP cores that capture and store values of signals within the FPGA.
 - Like an "embedded" logic analyzer.
 - Software tool that allows one to "read" the captured data and visualize it on a host computer.
 - Like the "screen" and "control panel" of a logic analyzer.

How It Works

- The designer connects signals of interest to a ChipScope ILA core.
 - ILA = Integrated Logic Analyzer.
- The designer connects the ILA to a ChipScope ICON core.
 - Allows a host computer to access the data stored in the ILA.
 - An ICON core is needed in ALL ChipScope designs!!!
- Test the system and use the ChipScope Analyzer tool to monitor the hardware signals connected to the ILA.
 - Setup trigger conditions.
 - i.e. "Begin to capture data when input3 goes high".
 - Plot captured data.

ChipScope System Setup



ChipScope Analyzer

🕲 Waveform - DEV:1	Му	Devi	ce1 (XC2	V6000) UI	NIT:0 MyIL	AO (ILA)								_ [X
Bus/Signal	х	0		10 80	120	160 I	200	240	280	320	360	400	440	480	
-DataPort[0]	1	1													
-DataPort[1]	0	0	ההתנתחת												
-DataPort[2]	1	1			ուսուս		MM				MM			ոոոու	
-DataPort[3]	0	0	תת	JUU						ЛЛ		UП			
-DataPort[4]	0	0													
-DataPort[5]	1	1													
-DataPort[6]	1	1													
-DataPort[7]	0	0													
-DataPort[8]	1	1													
-DataPort[9]	0	0													
-DataPort[10]	0	0													
- DotoDort[11]	1	1	4											Þ	
,					x	: 0	<u> </u>	• 0:	0	1	. Δ(x-0): 0	I		

Example: ChipScope IBA

- IBA = Integrated Bus Analyzer.
- A "special" version of an ILA tailored specifically for PLB and OPB CoreConnect busses.
- Allows one to passively monitor all bus traffic.
 - Address and data lines.
 - Bus control lines.
- Easy to use...

IBA: General Instructions

- Instantiate ChipScope cores:
 - Instantiate an ICON core.
 - Instantiate an OPB IBA core.
 - NOTE all ChipScope cores can be found in the IP catalog under "Debug".
- Connect ChipScope cores to the "system":
 - Connect the IBA core to the bus of interest.
 - Connect a control line from the ICON core to the IBA core.
- Configure the settings of the IBA/ILA.
 - Which signals to monitor, buffer-depth, etc.

Adding the ICON Core

- Open up a system to add ChipScope capabilities to...
- Go to the IP Catalog.
- Expand options for "Debug".
- Right-click on ICON.
- Click on "Add IP".



Adding the IBA Core

- Go to the IP Catalog.
- Expand options for "Debug".
- Right-click on OPB-IBA.
- Click on "Add IP".

	1		<u>×</u>		Filters
Project Applications IP C	atalog			O L L	😇 🕞 Bus I
 ● 				BBB	Name
Name V	Version	Description	<u> </u>		∲- ⊘ micn
Communication High-Speed			_		Ē- ⊘ mb_
Communication Low-Speed			1		🗉- 🗢 ilmb
₽-DMA					🖻 - 🍽 dimb
E-Debug					€-≪debu
• agilent_atc2	1.00.a	Agilent Trace Core			🗄- 🍽 dimb
•• • agilent_mtc_v4	1.00.a				🗄- 🗢 ilmb
 agilent_mtc_v5 	1.00.b	Agilent MicroBlaze v	5 Trace		€- ⊘ RS2
• chipscope_icon	1.01.a	Chipscope Integrated	d Contro		E- CED
• chipscope_ila	1.01.a	Chipscope Integrated	d Logic /		€- ⊃ DIPS
e chipscope_opb_iba	1.01.a	Chipscope OPB	orated P		ii-OP-st
chipscope_plb_iba	1.01.a	Chipscope PLB I	Add IP		pi
4			View MPD		w
×			View IP Mo	difications (Chang	e Log)
			View PDF (Datasheet	

Connect OPB-IBA to the OPB

	Filters				
O L L P M M	Bus Interface C Port	Connection Filters			
BBB	Name	Bus Connection	ІР Туре	IP Version	
	E- debug_module		opb_mdm	2.00.a	
	⊡-≪dimb_cntir		Imb_bram_if_cntir	2.00.a	
	E- Imb_cntir		Imb_bram_if_cntIr	2.00.a	
	B- RS232_Uart_1		opb_uartlite	1.00.b	
	₽- →LEDs_4Bit		opb_gpio	3.01.b	
	DIPSWs_4Bit		opb_gpio	3.01.b	
IT	€		opb_gpio	3.01.b	
	->chipscope_icon_0		chipscope_icon	1.00.a	
	□- chipscope_opb_iba_0		chipscope_opb_iba	a 1.00.a	
	MON_OPB	mb_opb			
-	€		opb_customcore	1.00.a	
	₽- ≫Imb_bram		bram_block	1.00.a	
🕩 System Ass	embly View1				

- Go to the System Assembly View.
 - Then go to the "Bus Interface" view.
- Expand options for the OPB-IBA.
- Now click on it's bus interface circle to connect it to the OPB bus.
 - The circle should go from hollow to shaded.

Connect a Control Line to IBA

•	Filters C Bus Interface @ Po	orts C Addresses	B	Filters (Ap	oplied)	4	Add Exte	mal Port		
Ν	ame	Net		Direction	Class	5	Sensitivity	Range	Frequency	Res
	control14	No Connection		0				[35:0]		
	control 15	No Connection		0				[35:0]		
ģ		_0								
	OPB_Clk	sys_clk_s		I	GLK					
	SYS_Rst	No Connection		1						
	chipscope_icon_c.	opb_iba_control		1				[35:0]		
	iba_trig_in	No Connection	opt	iba contr	ol			[C_G		
	iba_trig_out	No Connection	ope	0						
E	opb_customcore_0									
ķ	- Imb_bram									
ė										-
4										→
	System Assembly Vi	ew1								

- Go to the System Assembly View.
 - Then go to the "Ports" view.
- Expand options for the OPB-IBA.
- Now connect a signal to it's icon control line...
 - I called my signal"opb_iba_control".
- Now we must connect the other "end" to the ICON core.

Connect Control Line to ICON

€ Bus Interface € Port	s C Addresses	P.	Filters (Ap	oplied)	🚓 Add Exte	rnal Port		
Name	Net		Direction	Class	Sensitivity	Range	Frequency	Res
Chipscope_icon_0								
control0	opb_iba_control	1	0			[35:0]		
control1	No Connection		ob iba cor	trol		[35:0]		1
control2	No Connection	4	0	litor		[35:0]		
control3	No Connection		0			[35:0]		
control4	No Connection		0			[35:0]		
control5	No Connection		0			[35:0]		
control6	No Connection		0			[35:0]		
control7	No Connection		0			[35:0]		
control8	No Connection		0			[35:0]		
- control9	No Connection		0			[35:0]		<u>۔</u>
System Assembly View	v1							

- Go to the System Assembly View.
 - Then go to the "Ports" view.
- Expand options for the OPB-ICON.
- Now connect the previously created control signal to it's control0 line...
- Now a control line is connecting the IBA and the ICON cores.

Configuring the IBA Core

OLL	€ Bus Interface C Ports	C Addresses	Connection Filters	;	
BBB	Name	Bus Connection	IP Type	IP Version	-
	E- debug_module		opb_mdm	2.00.a	_
	⊡-≪dimb_cntir		Imb_bram_if_cntir	2.00.a	
	B- Imb_cntlr		Imb_bram_if_cntlr	2.00.a	
	B- RS232_Uart_1		opb_uartlite	1.00.b	
	E- CEDs_4Bit		opb_gpio	3.01.b	
	E- →DIPSWs_4Bit		opb_gpio	3.01.b	
III	PushButtons_5Bit		opb_gpio	3.01.b	
			chipscope_icon	1.00.a	
	Chipscope_opb_iba_0 B- Opb_customcore_0	Con	figure IP		
	🖯 🗢 Imb_bram	Viev	v MPD		
		View	v IP Modifications (Change Log)	1 3
C System Ass	embly View1	Viev	v PDF Datasheet	onungo zogy	
		Brow	wse HDL Sources		-
		Dele	ete Instance		

- Go to the System Assembly View.
 - Then go to the "Bus Interface" view.
- Right-click on the OPB-IBA core.
- Click on "Configure IP" to open up this cores configuration GUI.

Configuring the IBA Core

÷	chipscope_opb_iba_0 : chipscope_opb_iba	_v1_00_a
User System		HUL 7 Restore
OPB Control OPB Address	Number of Match Units for OPB Control	1 1
OPB Data	Matching Counter Width for OPB Control	0
OPB Write Data	Match Type for OPB Control Unit	asic with edges
OFB Protocol Violation Generic Trigger OFB Master OFB Stave Misc		
	Command Output Panel	
		<u>OK</u> ancel

- This GUI allows one to graphically configure the core's "generics".
 - Generics are compile-time parameters used to adjust the functionality of IP cores written in VHDL/Verilog.
- You can adjust what types of signals are monitored, buffer depth, etc. from here.

The Results

- EDK provides a GUI interface to build systems for platform FPGAs.
- The .mhs file is a textual representation of the components and connections that compose an EDK system.
- The ChipScope core instantiations should look like this in the .mhs...

```
BEGIN chipscope_icon
PARAMETER INSTANCE = chipscope_icon_0
PARAMETER HW_VER = 1.00.a
# Needed if opb_mdm is also present
PARAMETER C_SYSTEM_CONTAINS_MDM = 1
PARAMETER C_NUM_CONTROL_PORTS = 1
PORT control0 = opb_iba_control
END
```

```
BEGIN chipscope_opb_iba
PARAMETER INSTANCE = chipscope_opb_iba_0
PARAMETER HW_VER = 1.00.a
PARAMETER C_DATA_UNITS = 1
PARAMETER C_CONTROL_UNITS = 1
PARAMETER C_WRDATA_UNITS = 1
PARAMETER C_RDDATA_UNITS = 1
PARAMETER C_ADDR_UNITS = 1
BUS_INTERFACE MON_OPB = mb_opb
PORT OPB_Clk = sys_clk_s
PORT chipscope_icon_control = opb_iba_control
END
```

Using ChipScope

- The ChipScope Analyzer is the tool that you will actually use.
- It provides an interface...
 - To configure the ILA core.
 - Triggering setup.
 - To view captured data.
- This tools uses the JTAG interface to "talk" to the ICON core.
 - Where is the XUP's JTAG interface?

Using ChipScope Analyzer

- First, download the ChipScope-enabled bitstream to the FPGA.
- Make sure to be running a program that generates bus traffic on the OPB.
- Now launch the analyzer application...
 - We will now use the analyzer to look at the bus traffic as well as the OPB protocol.

Setting Up Analyzer For IBA

- First, click on the "Open Cable..." icon.
 - This opens up a JTAG connection to the ChipScope cores.
 - Click OK on any pop up dialogs.
- Now click on "File", "Import".
 - Select the IBA configuration (*.cdc) file from...
 - /implementation/<iba_core>/<iba_core>.cdc
 - This groups and names all of the bus signals.
- Now we can begin analyzing signals by using the trigger buttons...

Establish JTAG Connection



- First, establish a JTAG connection between host and FPGA.
 - Click on icon with black boxes in it.
- Click OK on subsequent dialog boxes.
- Now, Analyzer is connected to the ChipScope cores.

Import Configuration File

	Signal Import	×
Import File		
File:	chipscope_opb_iba_0.cdc	
Directory.	/users/jagron/388labs/chipscopeMB/implementation/chipscope_opb_iba_0_wrapper/	
	Select New File	
Unit/Device		
	DEV: 2 UNIT: 0 (IBA)	
	OK Cancel	

- The Xilinx toolset automatically produces a .cdc configuration file for IBA cores.
 - This file contains all of the grouping and naming conventions for the IBA signals.
- Go to "File", "Import", and select the .cdc file from...
 - /implementation/<iba_core>/<iba_core>.cdc

Triggering Setup

- Now, ChipScope is fully setup for use with the IBA core.
- To capture data immediately, click on the "T!" icon.
- To capture data based on custom trigger conditions...
 - First setup the proper Boolean triggering condition.
 - Then click on the sideways triangle icon to arm the trigger.
 - Now, wait for condition to occur.

Example: Immediate Trigger



Example: Custom Trigger (trigger when opb_select = '1')

New Project	🗱 Trigger Setup – DEV:2 MyDevice2 (XC2VP30) UNIT:0 MyIBA/OPB0 (BA/OPB)										
JTAG Chain	Match U	nit	Function		V	alue		Radix	Counter		
DEV:0 MyDevice0 (XCF32P) DEV:1 MyDevice1 (System AC	MO:TRIGO: OF	==			X_X000X_X	0(1,000(.)0	OOX Bin	disabled			
P-DEV:2 MyDevice2 (XC2VP30)											
UNIT:0 MyBA/OPB0 (IBA/	Add Activ	Active Trigger Condition		Name		Trigger Condi	tion Equation		Output Enable	e	
- Waveform	🖻 📃 🔍		TriggerConditio	on0		M	0		Disabled		
Licting								De sistema -	-		
Signals: DEV: 2 UNIT: 0	e Type: Window		lindows:		I Depth: 5:	12	•	Position:	5		
Data Port	Wayaform - DE	3/2 MyDevice	2 AC2VP30 UN	T-0 MyIRA/C						×	
	E mareronn - De	.v.z mybevice	2 ((021130) 011	-5	-4	-3 -7		<u></u>	1 2	4 1A	
OPB_DBUS	Bus/Signal	X	0	ō		- <u> </u>	1 	Ŭ			
OPB_ABUS	← OPB_RDDBUS	00000000	00000000			00000000			X 00000008X		
- CH: 0 SYS_Rst	← OPB_DBUS	00000000	00000000 -			00000000			X 00000008X		
- CH: 1 DEBUG_SYS_Rst	- OPB ABUS	40600008	00000000		0000000	γ		40600008	γ		
- CH: 2 WDT_Rst - CH: 3 OPB Rst		00000000						10000000		_	
- CH: 4 OPB_BE[3]	CLAPPERMODO2							U		_	
- CH: 5 OPB_BE[2] - CH: 6 OPB_BE[1]	- OPB_BE [3]	1	° –								
- CH: 7 OPB_BE[0]	- OPB_BE [2]	1	° _								
- CH: 8 OPB_select	- OPB_BE [1]	1	0								
- CH: 10 OPB_RNW	- OPB_BE [0]	1	0								
- CH: 11 OPB_errAck	- OPB_select	1	0								
- CH: 13 OPB_toutSup	- OPB yferåck	0									
- CH: 14 OPB_retry											
- CH: 15 OPB_seqAddr - CH: 16 OPB_busLock	UPB_KNW	1	-								
- CH: 17 OPB_ABus[31]	- OPB_errAck	0	° –								
- CH: 18 OPB_ABUS[30] - CH: 19 OPB_ABUS[29]	- OPB_timeout	0	<u> </u>								
- CH: 20 OPB_ABus[28]			4 • •								
- CH: 21 OPB_ABus[27] - CH: 22 OPB_ABus[26]					X: 129	 	: -5	A(>	(-0): 134		
- CH: 23 OPB_ABus[25]	L									_	
- CH: 24 OPB_ABus[24]											

Project

- Create a ChipScope-enabled hardware system with IBA.
- Test the setup using a software application.
- Use ChipScope Analyzer to capture the following situation.
 - Read a value from an address.
 - Write the (value + 1) to (address + offset).
- NOTE address offsets of are meant to be word-aligned!!!!

Example Results



ChipScope ILA Setup

- ILA setup is the same as for IBA except...
 - ILA can monitor arbitrary signals.
 - One must connect up each signal that needs to be monitored to the ILA unit.
- For further help...
 - ChipScope Online Documentation.
 - EECS 388 Wiki Documentation.
 - TA Assistance.