Software Driven Test of FPGA Prototype Methods & Use cases

Krzysztof Szczur, Aldec Inc. Radosław Nawrot, Aldec Inc.

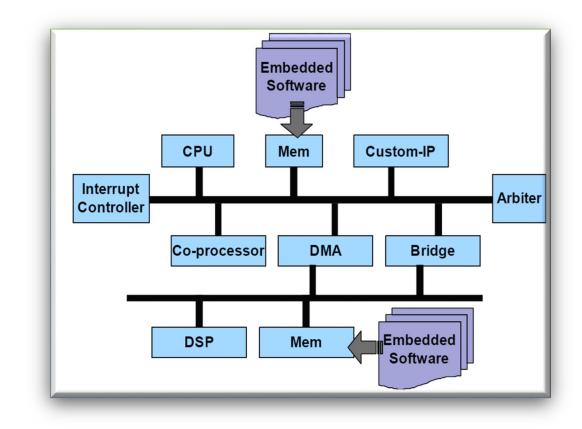






Motivation - SoC verification challenge

- SoC = Hardware + Software
 - both need verification
 - software on critical-path (schedule risk)
- Complex architecture
 - Hierarchical Network-on-Chip (NoC)
 - Corner case hardware/software bugs
 - Re-usable IPs: CPU, MEM, DMA, ...
 - RTL code not available for FPGA prototyping
 - Custom IPs: Co-processors, Function accelerators
 - Often need late hardware debugging







Tutorial Agenda

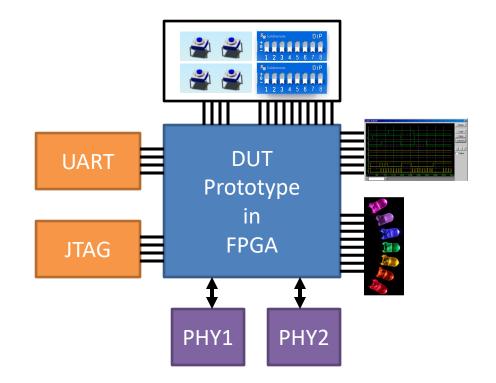
- Software driven v.s. traditional
- FPGA Technology enabling software driven verification
- Embedded software driven testbench application examples with HES-US-440
 - Using HES Proto-AXI host interface
 - Bare metal approach
- Software driven testbench in simulation
 - QEMU co-simulation with Riviera-PRO





Traditional Prototyping Bench

- Controllability & Configuration
 - Dip switches
 - Push buttons
 - UART or JTAG ports
- Status & Debug
 - LED & Panels
 - Logic analyzers
- Live data streams
 - PHY components
 - Peripheral devices



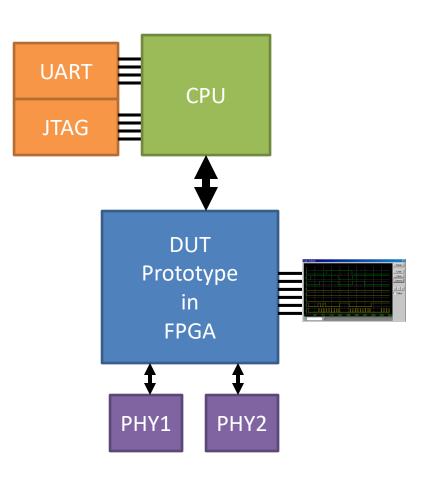




Software Driven Testbench (SDT)

- Controllability & Configuration

 Use CPU
- Status & Debug
 - Use CPU
 - Logic analyzers (external & on-chip)
- Live data streams
 - PHY components
 - Peripheral devices



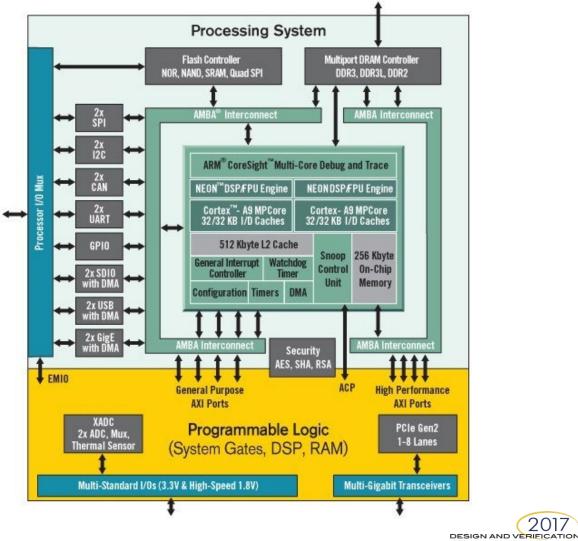




FPGA Technology Enabling SDT

Xilinx Zyng and the like...

- Processing System
 - ARM Cortex CPU
 - Standard I/O peripherals
- Programmable Logic
 - Classic FPGA used for:
 - More peripherals
 - Glue logic & Bridge



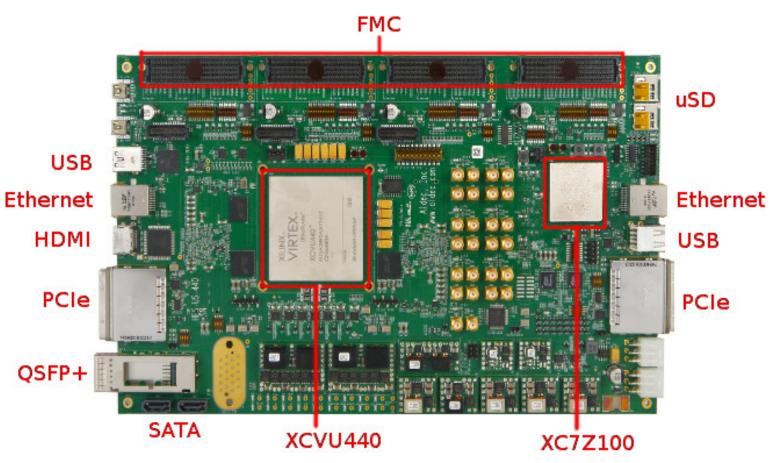
2017

ONFERENCE AND EXHIBITION



Aldec HES-US-440

- Main FPGA:
 - Virtex UltraScale™ XCVU44
 - 26 Million ASIC Gates
- Host FPGA
 - Zynq-7000, XC7Z100
 - CPU: ARM Cortex A9
 - FPGA: Kintex-7





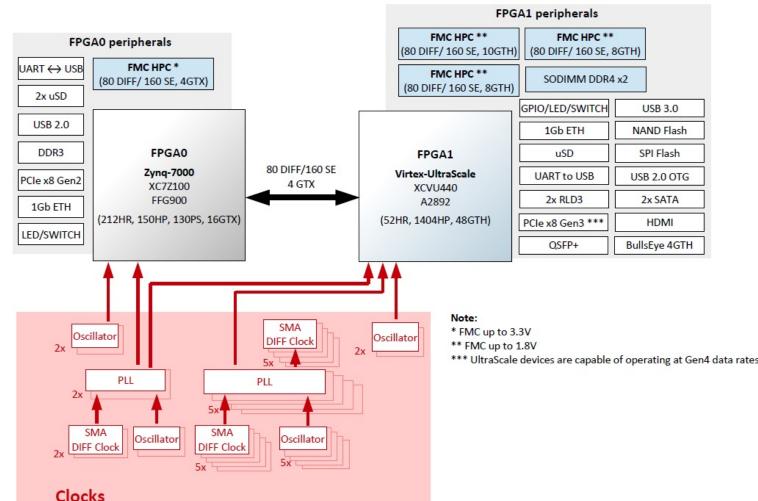
HES-US-440 Diagram

• Memory

- DDR4 (64 GB) : 2 x SO-DIMM
- RLDRAM-3 (1152 Mb)
- NAND Flash, SPI Flash
- 2x MircoSD card slots

• I/O Connectivity

- Inter-FPGA:
 - 80 DIFF / 160 SE, 4 GTH
- FMC connectors
 - 320 DIFF / 640 SE, 26 GTH
- Peripherals
 - PCIe x8 gen3 & gen2
 - USB 3.0 & 2.0 & OTG, SATA
 - Ethernet 1Gb, QSFP+ 40Gb
 - BullsEye
- Clocks
 - 7x programmable clock modules

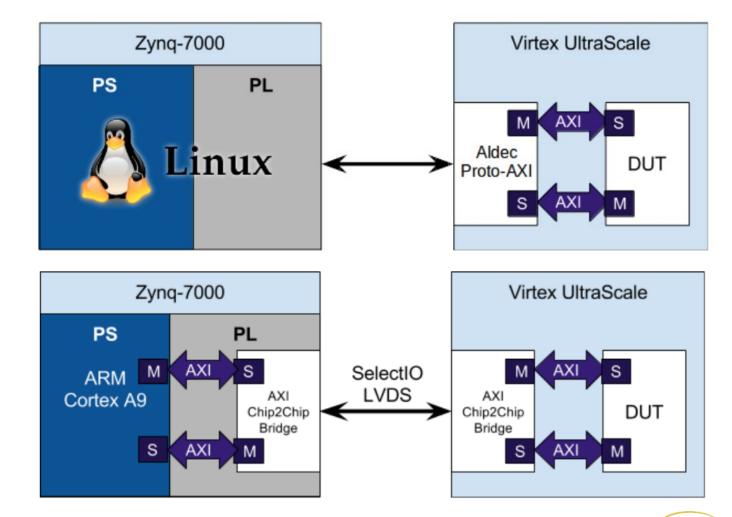






SDT application examples on HES-US-440

- Case-1: Aldec HES-ProtoAXI
 - Zynq image ready to use (PL & PS)
 - Use Aldec Proto-AXI IP
 - Ready Embedded Linux env.
 - Simple Proto-AXI C/C++ API
 - Spend all your time to develop SDT
- Case-2: Bare metal / Custom
 - Use Xilinx Vivado & SDK
 - Create custom C2C bridge in PL
 - Create custom SDT in PS

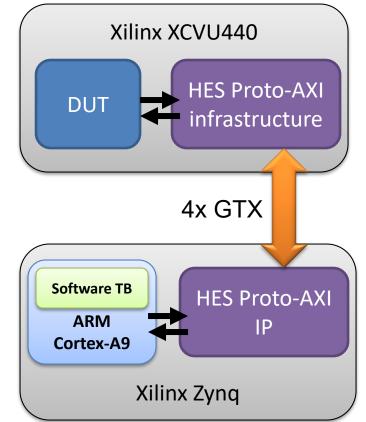


ESIGN AND VERIF



Use Case-1 with HES Proto-AXI

- 1. Connect DUT to HES Proto-AXI & implement FPGA bitstream
- 2. Develop & build software TB
- 3. Run test

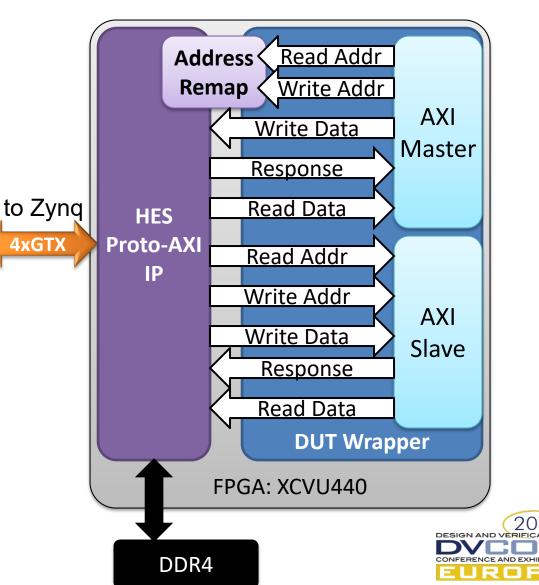






Step-1: Connecting DUT to HES-Proto-AXI

- Connect HES Proto-AXI
 - Use any HDL Editor
 - VHDL or Verilog templates available
 - Fixed size AXI4 memory mapped interface
 - Master interface is optional
- Configure address translation parameters (Address Remapper)
- DUT gets access to on-board DDR4 through HES Proto-AXI
- Compile and implement project in Xilinx Vivado





Step-2: Developing & building testbench

- Develop testbench as C/C++ code
 - Use HES Proto-AXI C API
 - #include "hesprotoaxiapi.h"
 - Simple, user app level API
 - needn't know Zynq or Linux kernel
- Compile and build
 - Cross-compilation on host PC
 - Use GCC toolchain from Xilinx SDK
 - arm-xilinx-linux-gnueabi-g++

```
// AXI write (blocking transaction)
```

HesProtoAxiTrans HesProt	oAxiWrite(
HesProtoAxiHandler	_handler,
HesProtoAxiLocation	_location,
HesProtoAxiU64	_address32,
HesProtoAxiBuffer	_buffer,
HesProtoAxiU64	_bufferSize32);

- // AXI read (non blocking transaction)
 bool HesProtoAxiReadNoCheck(
 - HesProtoAxiHandler _handler, HesProtoAxiLocation _location, HesProtoAxiU64 _address32, HesProtoAxiBuffer _buffer, HesProtoAxiU64 _bufferSize32);

// GPI write

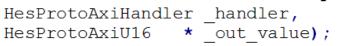
```
HesProtoAxiStatus HesProtoAxiWriteGpi(
HesProtoAxiHandler _handler,
HesProtoAxiU16 _value);
```

// GPIO read

HesProtoAxiStatus HesProtoAxiReadGpio(

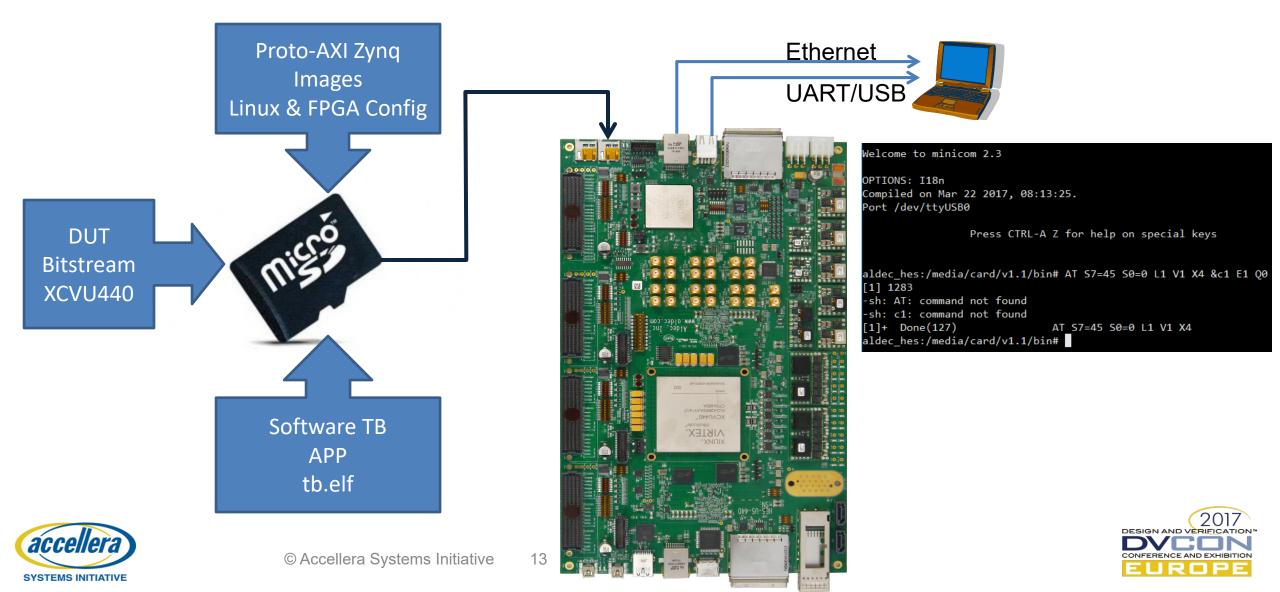


12





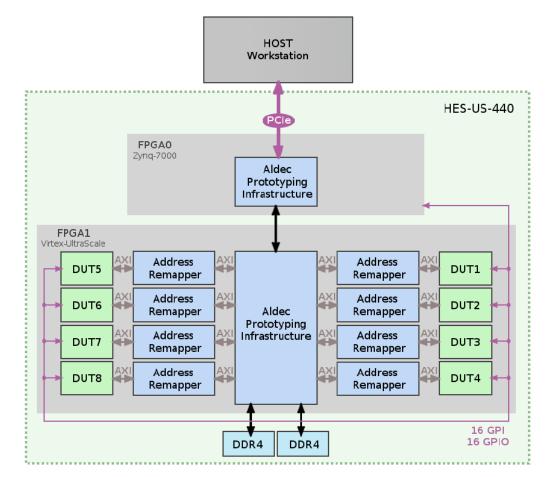
Step-3: Running Test



HES Proto-AXI Highlights

Data for HES-US-440

- AMBA AXI4 memory mapped
- 8x Master & Slave interfaces
- Data width: 256, Burst length: 256
- Local clock: 160 MHz
- RAM space: 16GB DDR4 + 64MB RLD
- Customizable address translation
- 16x GPI and GPIO additional lines
- Simple C API
- HES Proto-AXI simulation model available
- Embedded & PC Host modes



ESIGN AND VERIF



Questions

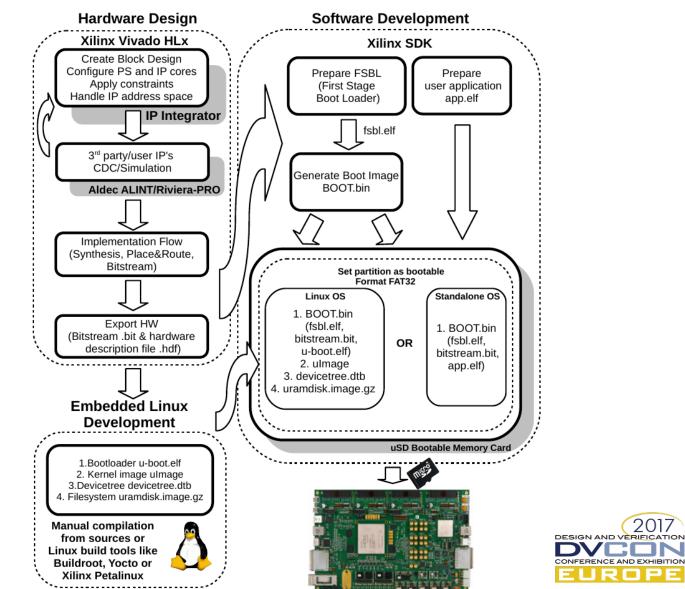
Next: Case-2: Bare metal approach





Use Case-2 – Bare metal approach

- 1. Test controller (Zyng)
 - A. Hardware design
 - Embedded Linux development Β.
 - C. Software development
- 2. Design connectivity (U440)
 - A. Custom interconnect
 - **FPGA** implementation Β.



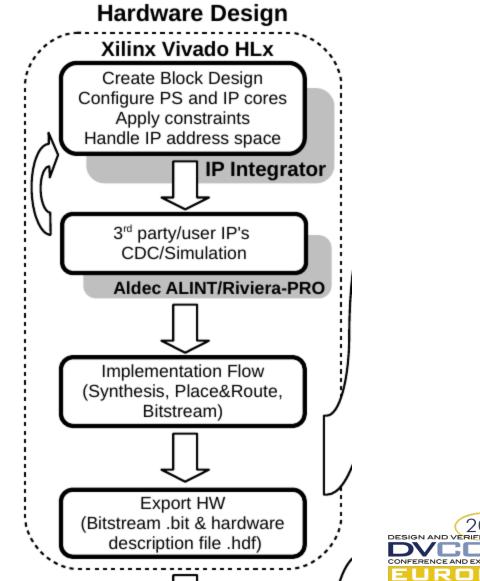
2017



© Accellera Systems Initiative

Step 1.A: Hardware design

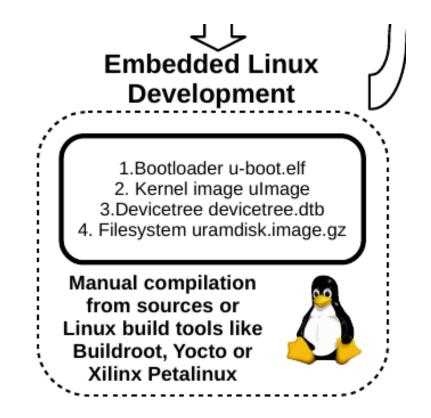
- Short path: start with reference design from Aldec
- Customize reference design
 - Add your custom chip-2-chip connectivity
- Use Aldec Riviera-PRO with QEMU to verify your changes
- Run Vivado to implement bitfiles and export hardware description





Step 1.B: Embedded Linux development

- Embedded Linux projects available:
 - Yocto
 - Petalinux
- Other than Linux:
 - Free RTOS
 - Bare metal no OS
 - Xilinx SDK provides toolchain for such mode



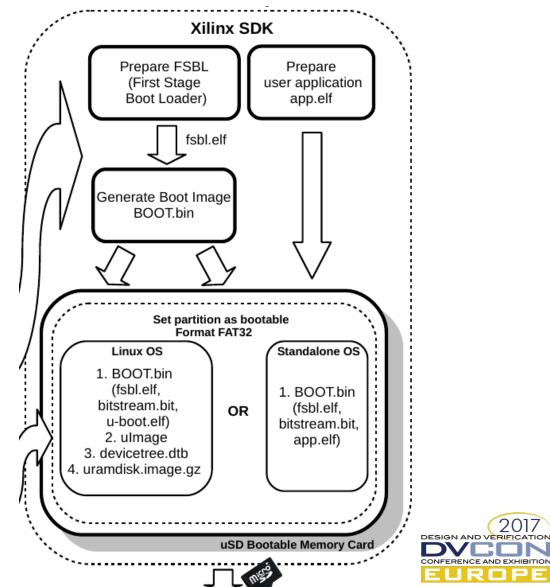




Step 1.C: Software development

19

- Use Xilinx SDK
 - to develop test application
 - create bootable Linux image
- Copy files to Micro-SD
 - Bitstream for FPGA (PL configuration)
 - Bootloader for ARM (PS configuration)
 - Linux image & device tree
 - User application (app.elf)

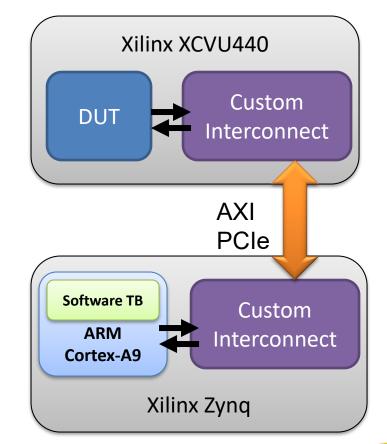


 20°



Step 2: Design connectivity

- Use Xilinx Vivado to implement XCVU440
 - Create custom interconnect
 - Coupling with one created in Zynq
 - Use any standard (AXI, PCIe)
 - Use any signaling (LVDS, GTX)
 - Connect with the DUT
 - Run FPGA implementation





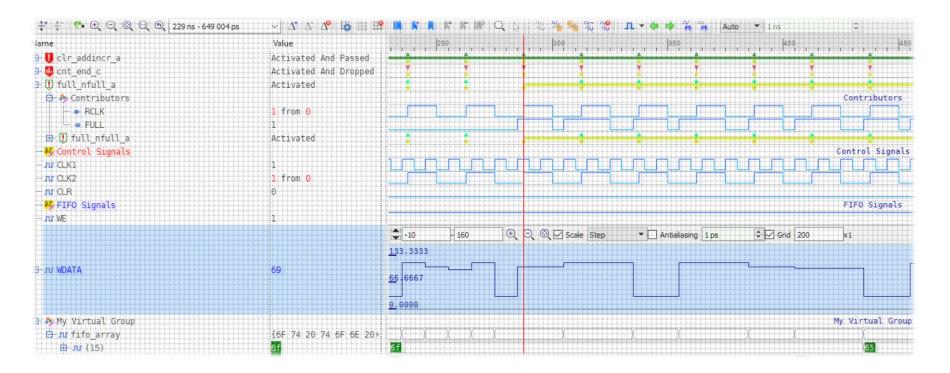


Questions

Next: Software driven testbench in simulation: QEMU co-simulation with Riviera-PRO







QEMU CO-SIMULATION WITH RIVIERA-PRO

Radosław Nawrot

Software Products Manager, Aldec Inc.



© Accellera Systems Initiative



QEMU co-simulation platform

- Parts of the flow:
 - Riviera-PRO Advance Verification Platform
 - -Aldec AXI BFM
 - -Aldec QEMU Bridge
 - -QEMU Emulator





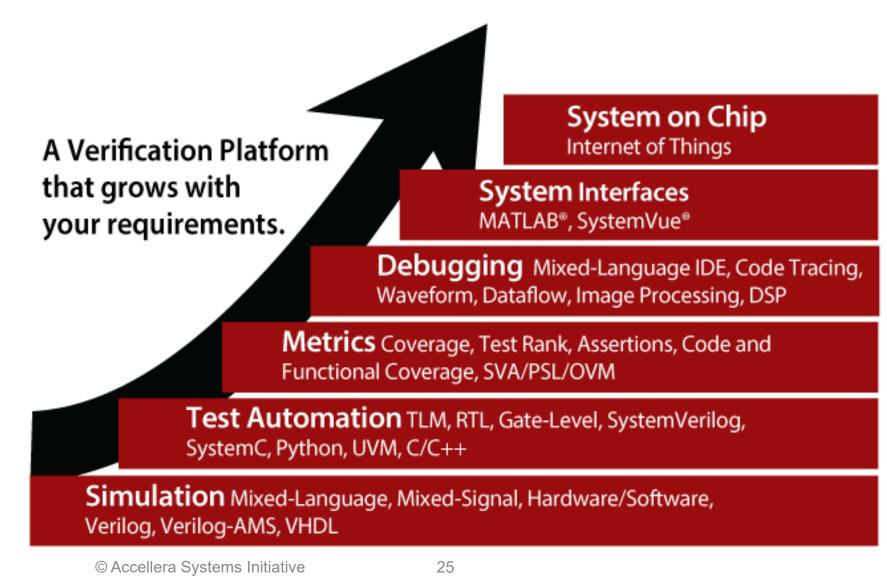


QEMU co-simulation platform

- Full debug capabilities of RTL IP Core in Riviera-PRO simulator:
 - Waveforms
 - Hardware Breakpoints
 - Hardware steps
 - Transaction based verification and debug
- Kernel and driver debug via GDB
 - Software Breakpoints
 - Variable probing
- Zynq Linux OS ready to use on QEMU without modifications



Riviera-PRO Verification Platform







Riviera-PRO Highlights

- High Performance Simulation
 - Extensive simulation optimization algorithms
 - Support for latest Verification Libraries: UVM, OSVVM , UVVM, CocoTB and more
- Advanced Debugging
 - Transaction Level simulation end debug
 - Multi-language debug environment (Verilog, VHDL, SystemVerilog, SystemC, Verilog-AMS)
 - Support for MATLAB and Simulink
 - C/C++ debug environment
 - Support for external C/C++ compilers (GCC, Visual C++)
 - UVM Toolbox, Graph and Class Viewer
 - Code tracing, Waveform, Dataflow, FSM window, Coverage, assertions, memory visualization
 - Comprehensive Assertions-Based Verification (SVA and PSL)
 - Advanced Code and Functional Coverage
 - User-defined test plan linking with coverage database
 - Plot Viewer and Image Viewer





ESIGN AND VER

QEMU – Virtual Platform

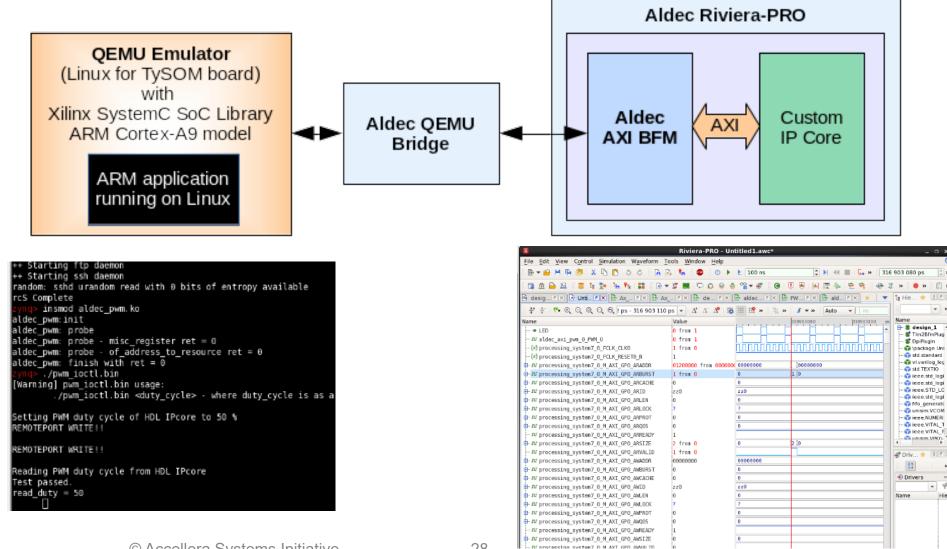
- Free and open-source
- Fast emulator
- Supports many machines:
 - Xilinx-zynq-a9
- Supports many ARM processors:
 - ARM926
 - ARM946
 - Cortex-A8
 - Cortex-A9
 - Cortex-A15
 - Cortex-A53
 - Cortex-A57







Riviera-PRO and QEMU



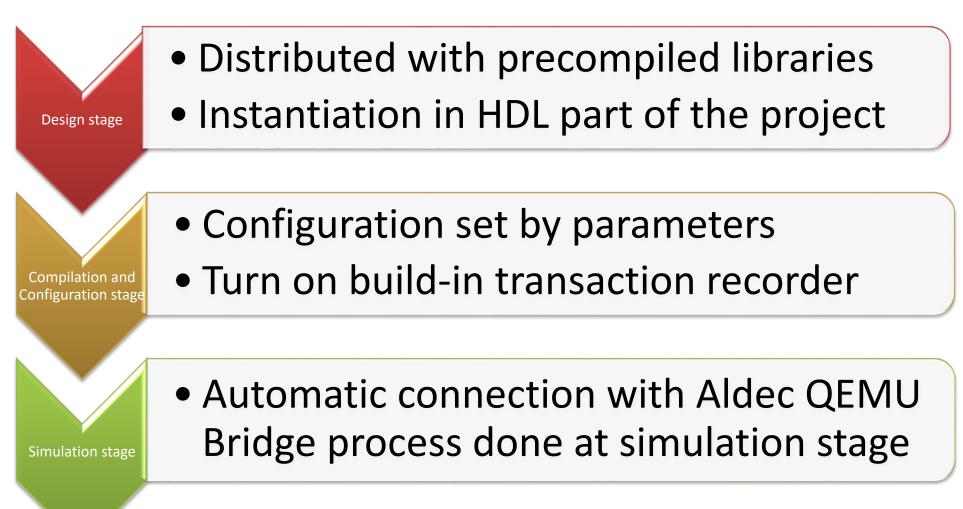




SYSTEMS INITIATIVE

28

Aldec AXI BFM usage





Aldec AXI BFM

- AXI 3 Master
- AXI 3 Slave
- AXI 4 Master
- AXI 4 Slave
- AXI 4 Lite Master
- AXI 4 Lite Slave
- AXI 4 Stream Master
- AXI 4 Stream Slave

1 Libraries		
0 🔢 🖧 88		- <i>S</i>
Name	Туре	Path
👜 👬 act3_ver (RO)		/home/radekn/Aldec/Riviera-PRO-2017.06-x 🔺
🕀 👖 aldec (RO)		/home/radekn/Aldec/Riviera-PRO-2017.06-x
🛱 👖 aldec_axi_bfm_qemu (RO)		/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 \$root	Module	
Ax_Axi3MasterBFM Name: aldec_ax	_bfm_qemu	
🕼 Ax_Axi3MasterBFM	apping	era-PRO-2017.06-x64/vlib/aldec_axi_bfm_qemu/aldec_axi_bfm_qemu.
Ax_Axi3SlaveBFM	adekn/Aldec/Rivie	era-PRO-2017.06-x64/viib/aidec_axi_bim_demu/aidec_axi_bim_demu.
🕾 Ax_Axi3SlaveBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 Ax_Axi4LiteMasterBFM	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 Ax_Axi4LiteMasterBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 Ax_Axi4LiteSlaveBFM	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 Ax_Axi4LiteSlaveBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 Ax Axi4MasterBFM	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 Ax_Axi4MasterBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 Ax_Axi4SlaveBFM	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 Ax Axi4SlaveBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 Ax_Axi4StreamMasterBFM	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 Ax_Axi4StreamMasterBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 Ax_Axi4StreamSlaveBFM	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🖅 Ax_Axi4StreamSlaveBFMcore	Module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🌇 TransactionRecorderAxi3	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🥵 TransactionRecorderAxi4	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🦝 TransactionRecorderAxi4Lite	Top module	/home/radekn/Aldec/Riviera-PRO-2017.06-x
🜍 UnitScopePackage_1	SV Package	/home/radekn/Aldec/Riviera-PRO-2017.06-x
8 protected_512eb4b7af147c73dbdfe0>		/home/radekn/Aldec/Riviera-PRO-2017.06-x
protected_ef16c1876adf1a8fe7b4971		/home/radekn/Aldec/Riviera-PRO-2017.06-x





AXI BFM instance in HDL Code

		<u>−</u>					
5	.AWVALID(processing system7 0 M AXI GP1 AWVALID),	« ***		Name	Design unit	Library	
6	.BID(processing system7 0 M AXI GP1 BID),		-	🚊 書 design 1	design 1	xil defaultlib	
57	.BREADY(processing system7 0 M AXI GP1 BREADY),			🖶 書 axi bram ctrl 0	design 1 axi bram ctrl	0⊩xil defaultlib	
8	.BRESP(processing system7 0 M AXI GP1 BRESP),			🕀 婁 axi bram ctrl 0 bram	design 1 axi bram ctrl	0⊧xil defaultlib	
9	.BVALID(processing system7 0 M AXI GP1 BVALID),			🕀 🔹 axi cdma 0	design 1 axi cdma 0 0(d xil defaultlib	
0	.RDATA(processing system7 0 M AXI GP1 RDATA),			🕀 書 axi_mem_intercon_1	design 1 axi mem inter	c⊧xil_defaultlib	
1	.RID(processing system7 0 M AXI GP1 RID),			🕀 書 axi protocol converter 0	design 1 axi protocol co	o⊮ xil defaultlib	
2	.RLAST(processing system7 0 M AXI GP1 RLAST),			🕀 🚦 master 0	Ax Axi3MasterBFM	aldec axi bfm gem	nu
3	.RREADY(processing system7 0 M AXI GP1 RREADY),			🕀 📲 master 1	Ax Axi3MasterBFM	aldec axi bfm gem	nu
4	.RRESP(processing system7 0 M AXI GP1 RRESP),			🕀 🛊 slave 0	Ax Axi3SlaveBFM	aldec axi bfm gem	nu
5	.RVALID(processing system7 0 M AXI GP1 RVALID),			🗘 @INITIAL#829_0@		xil_defaultlib	
6	.WDATA(processing system7 0 M AXI GP1 WDATA),			🙀 @INITIAL#841_1@		xil_defaultlib	
7	.WID(processing system7 0 M AXI GP1 WID),			🖶 📲 gibi	glbl	xil_defaultlib	
8	.WLAST(processing system7 0 M AXI GP1 WLAST),			📽 TIm2BfmPlugin	TIm2BfmPlugin	aldec_axi_bfm_qem	nu
9	.WREADY(processing system7 0 M AXI GP1 WREADY),			📽 DpiPlugin	DpiPlugin	aldec_axi_bfm_qem	nu
Θ	.WSTRB(processing system7 0 M AXI GP1 WSTRB),			🛶 🎲 \package UnitScopePackage_1\	UnitScopePackage_1	aldec_axi_bfm_qem	nu
1	.WVALID(processing_system7_0_M_AXI_GP1_WVALID)			🌍 std.standard	standard	std	
2 -);		:	Name: std.standard	verilog_logic	vl	
3				Hierarchy Path: sim:/	TEXTIO	std	
4 ▷ 🗐	Ax Axi3SlaveBFM #(🐨 🕫 Type: Package	std_logic_1164	ieee	
5	DATA BUS WIDTH(64),			🐨 🌍 ie Language: VHDL	NUMERIC_STD	ieee	
6	.ADDRESS WIDTH(32),			a Design Unit: standard ram_ctrl_funcs	axi_bram_ctrl_funcs	axi_bram_ctrl_v4_0	_10
7 -	.ID WIDTH(12))			ず ie Library: std	std_logic_arith	ieee	
8 😐	slave 0 ⁻ (🗝 🧊 unisim.VCOMPONENTS	VCOMPONENTS	unisim	
9	.ACLK(processing system7 0 FCLK CLK0),			🗝 🌍 ieee.VITAL_Timing	VITAL_Timing	ieee	
Э	.ARESETn(processing system7 0 FCLK RESET0 N),			🕆 🌍 ieee.VITAL_Primitives	VITAL_Primitives	ieee	
1	.ARADDR(axi mem intercon 1 M01 AXI ARADDR),			🗝 🎲 unisim.VPKG	VPKG	unisim	
2	.ARBURST(axi mem intercon 1 M01 AXI ARBURST),			🗝 🌍 ieee.STD_LOGIC_UNSIGNED	STD_LOGIC_UNSIGNED	ieee	
3	.ARCACHE(axi mem intercon 1 M01 AXI ARCACHE),			🗠 🕿 ieee std. Ionic. misc	std logic misc	ieee	
4	.ARID({1'b0,1'b0,1'b0,axi mem intercon 1 M01 AXI ARID}),			Hierarchy Datasets Classes			
5	.ARLEN({4'b0,axi mem intercon 1 M01 AXI ARLEN}),						
6	.ARLOCK(axi mem intercon 1 M01 AXI ARLOCK),		1	Objects			
7	.ARPROT(axi_mem_intercon_1_M01_AXI_ARPROT),						
в	.ARREADY(axi_mem_intercon_1_M01_AXI_ARREADY),		-				
9	.ARSIZE(axi_mem_intercon_1_M01_AXI_ARSIZE),			Name		lue	Ту
Э	.ARVALID(axi_mem_intercon_1_M01_AXI_ARVALID),			I ACLK	1		wi
1	.AWADDR(axi_mem_intercon_1_M01_AXI_AWADDR),			I ARESETN	1		wi
2	.AWBURST(axi_mem_intercon_1_M01_AXI_AWBURST),			🖶 🕨 AWID	z?()	[10



20

DESIGN AND VERIFICATION

CONFERENCE AND EXHIBITION

AXI BFM instance in HDL Code

<u>W</u> indow	/ <u>H</u> elp										
9 88	500	🕐 🕨 🛓 100 ns 🗘 🕨 📢 🔳 🖡 🖧 📮 📫 1 372 ms	¢ 4	2	20						
; .	I 🖓 🗘	😔 🕹 😭 🛪 🐨 🔞 I 🛡 I 🕪 🔚 😓 I 🚖 🗣 🐵 🗷 🗵 🔴 🗣 🌒 🗣 🗮	// 01•A	1	, 🖄 🕲 🛑						
👌 desig	gn_wrapp	er_compile.do 🛪 🖾 Untitled1.awc* 🛪 🗋 DpiBridge.so रू 🎼 design_1.v र 🛪 🔹	•	L	Hierarchy						± \$₹
88		· · · · · · · · · · · · · · · · · · ·	»						•	🖗 (.*)	Attributes
682	11	.S AXI HPO WDATA(axi mem intercon 1 MO1 AXI WDATA),	«	Ν	lame	Design unit		Library			
683	11	.S AXI HPO WID({1'b0,1'b0,1'b0,1'b0,axi mem intercon 1 M01 AXI WID})			🕂 書 design_1	design_1		xil_defaultlib			
684	11	.S AXI HPO WLAST(axi mem intercon 1 M01 AXI WLAST),			🖶 🖶 axi_bram_ctrl_0	design_1_axi_bram_ct	rl_0⊧	xil_defaultlib	1		
685	11	.S AXI HPO WREADY(axi mem intercon 1 M01 AXI WREADY),			🖶 書 axi_bram_ctrl_0_bram	design_1_axi_bram_ct	rl_0⊧	xil_defaultlib			
686	11	.S AXI HPO WRISSUECAPI EN(1'b0),			🖶 書 axi_cdma_0	design_1_axi_cdma_0	_0(d)	xil_defaultlib			
687	11	.S AXI HPO WSTRB(axi mem intercon 1 M01 AXI WSTRB),			🖶 書 axi_mem_intercon_1	design_1_axi_mem_in	terc⊧	xil_defaultlib			
688	-77	.S AXI HPO WVALID(axi mem intercon 1 M01 AXI WVALID));			🖶 書 axi_protocol_converter_0	design_1_axi_protocol	co⊮	xil_defaultlib			
689	1				🖶 書 master_0	Ax Axi3MasterBFM	-	aldec_axi_bfm_qemu	-		
690	/	/assign processing system7 0 M AXI GP0 WID = 12'b0;			🕀 書 master 1	Ax Axi3MasterBFM		aldec axi bfm qemu			
		x Axi3MasterBFM #(🖶 書 slave 0	Ax Axi3SlaveBFM		aldec axi bfm qemu			
692	Ī	.DATA BUS WIDTH(32),			- 🗘 @INITIAL#829 0@	_		xil_defaultlib			
693		.ADDRESS WIDTH(32),						xil defaultlib			
694	L	.ID WIDTH(12))			🗄 📲 gibi	glbl		xil defaultlib			
	E m	haster 0 (📽 TIm2BfmPlugin	- TIm2BfmPlugin		aldec axi bfm gemu			
696	ī "	.ACLK(processing system7 0 FCLK CLK0),			- PpiPlugin	DpiPlugin		aldec axi bfm gemu			
697		.ARESETn(processing_system7_0_FCLK_RESET0_N),			\package UnitScopePackage 1\	UnitScopePackage 1		aldec axi bfm gemu			
698		.ARADDR(processing_system7 0 M AXI GP0 ARADDR),			std.standard	standard		std			
699		.ARBURST(processing_system7_0_M_AXI_GPO_ARBURST),				verilog logic		vl			
700		.ARCACHE(processing_system/ 0 M AXI GPO ARCACHE),			std.TEXTIO	TEXTIO		std			
701		.ARID(processing system7 0 M AXI GP0 ARID),			ieee.std logic 1164	std logic 1164		ieee			
702		.ARLEN(processing_system/_0_H_ARL_OF0_ARLEN),			ieee.NUMERIC STD	NUMERIC STD		ieee			
703		.ARLOCK(processing system7 0 M AXI GP0 ARLOCK),			axi bram ctrl v4 0 10.axi bram ctrl funcs	axi bram ctrl funcs		axi bram ctrl v4 0 10			
704		.ARPROT(processing_system7_0_M_AXI_GP0_ARPROT),			ieee.std logic arith	std logic arith		ieee			
705		.ARREADY(processing_system/ 0 M AXI GPO ARREADY),			unisim.VCOMPONENTS	VCOMPONENTS		unisim			
706		.ARSIZE(processing system7 0 M AXI GPO ARSIZE),			ieee.VITAL Timing	VITAL Timing		ieee			
707		.ARVALID(processing_system/ 0 M AXI GPO ARVALID),			ieee.VITAL Primitives	VITAL Primitives		ieee			
708		.AWADDR(processing_system7_0_M_AXI_GPO_AWADDR),			unisim.VPKG	VPKG		unisim			
709		.AWBURST(processing system7 0 M AXI GPO AWBURST),			ieee.STD LOGIC UNSIGNED	STD LOGIC UNSIGNED		ieee			
710						std logic misc		ieee			
		.AWCACHE(processing_system7_0_M_AXI_GP0_AWCACHE),			Hierarchy Datasets Classes			·····			
711		.AWID(processing_system7_0_M_AXI_GP0_AWID),		5							
712		.AWLEN(processing_system7_0_M_AXI_GP0_AWLEN),			• Objects						📩 ≑ ह
713		.AWLOCK(processing_system7_0_M_AXI_GP0_AWLOCK),									
714		.AWPROT(processing_system7_0_M_AXI_GP0_AWPROT),			I				•	🌳 (.*)	Attributes
715		.AWREADY(processing_system7_0_M_AXI_GP0_AWREADY),	-	N	Jame		Value	е Туре	<u>م</u>		
716		.AWSIZE(processing_system7_0_M_AXI_GP0_AWSIZE),			anne axi bram ctrl 0 BRAM PORTA ADDR		No da		e D]wire		
717		.AWVALID(processing_system7_0_M_AXI_GP0_AWVALID),			axi bram ctrl 0 BRAM PORTA CLK		No da No da				
718		.BID(processing_system7_0_M_AXI_GP0_BID),			axi_bram_ctrl_0_BRAM_PORTA_CLK		No da No da		0]wire		
719		.BREADY(processing_system7_0_M_AXI_GP0_BREADY),	Ŧ		Objects Drivers/Readers		wo da	1.a [31:0	Jwire		



DESIGN AND VERIFICA

Built-in Transaction Recorder

🌛 desigi	n_wrapper_com 🖲 🗙 📴 Untitled 🖲 🗙 🗋 DpiBri Ə 🗙 🗟 desi Ə 🗙 🔂 Ax_Axi3Slav Ə 🗙 🔶	•	Ľ	Hierarchy					5\$
88	▼ # # !♥ # # # # !♥ !♥ !♥ !♥ !■ ■ ■ ■ ■ ■ ■	»					•	🌳 (.*)	Attributes
704	axi_slv_bfm_core.wait_on(phase, count);	« ***		lame	Design unit	Library			
705	endtask	•	E		design_1	xil_defaultlib			
706	//////////////////////////////////////			🖶 🖶 axi_bram_ctrl_0	design_1_axi_bram_ctr				
707	`endif				design_1_axi_bram_ctr				
708					design_1_axi_cdma_0_				
709					design_1_axi_mem_inte				
	`ifdef ALDEC_USE_TRANSACTIONS				design_1_axi_protocol_				
711				🖶 🖶 master_0	Ax_Axi3MasterBFM	aldec_axi_bfm_qemu			
712 🖃				🖶 🛊 master_1	Ax_Axi3MasterBFM	aldec_axi_bfm_qemu			
713	.DATA_BUS_WIDTH(DATA_BUS_WIDTH),			🖻 🛊 slave_0	Ax_Axi3SlaveBFM	aldec_axi_bfm_qemu			
714	.ADDRESS_WIDTH(ADDRESS_WIDTH),			axi_slv_bfm_core	Ax_Axi3SlaveBFMcore	aldec_axi_bfm_qemu			
715	.ID_WIDTH(ID_WIDTH),			TransRecorder	TransactionRecorderAx				
716	.INSTANCE(\$sformatf("%m")))			└──☆ @INITIAL#ini@		aldec_axi_bfm_qemu xil defaultlib			
717 Ξ				@INITIAL#829_0@ @INITIAL#841 1@		xil_defaultlib			
718	ACLK(ACLK),	11	Π,		glbl	xil_defaultlib			
719	. ARESETN (ARESETN),			Tim2BfmPlugin	gibi Tim2BfmPlugin	aldec axi bfm gemu			
720	. AWID (AWID),				DpiPlugin	aldec_axi_bfm_gemu			
721 722	. AWADDR (AWADDR),			package UnitScopePackage 1	UnitScopePackage 1	aldec_axi_bfm_qemu			
723	.AWLEN(AWLEN), .AWSIZE(AWSIZE),			std.standard	standard	std			
723	.AWBURST(AWBURST).			vl.verilog logic	verilog logic	vl			
725	.AWDORST (AWDORST),			std.TEXTIO	TEXTIO	std			
726	. AWCACHE (AWCACHE) ,			ieee.std logic 1164	std logic 1164	ieee			
727	.AWCACHE (AWCACHE),			ieee.NUMERIC STD	NUMERIC STD	ieee			
728	.AWVALID(AWVALID).			-	axi bram ctrl funcs	axi bram ctrl v4 0 1	10		
729	. AWREADY (AWREADY) ,			ieee.std logic arith	std_logic_arith	ieee			
730	.WID(WID),			unisim.VCOMPONENTS	VCOMPONENTS	unisim			
731	WDATA(WDATA),				VITAL Timing	ieee			
732	.wstrb(wstrb),			ieee VITAI Primitives	VITAI Primitives	ieee			1
733	.WLAST(WLAST),			Hierarchy Datasets Classes					
734	.WVALID(WVALID),								
735	.WREADY (WREADY),		n,	• Objects					📩 ≑
736	.BID(BID),						-	😪 (.*)	Attributes
737	.BRESP(BRESP),			۰ -					
738	.BVALID(BVALID),		N	lame			Туре		
739	.BREADY(BREADY),			ACLK	1		wire		
740	.ARID(ARID),			ARESETn	1		wire		
741	. ARADDR (ARADDR) ,	-	E	🖶 🕨 AWID	z	?0	[ID_WIDTH-1:0]w	ire	



DESIGN AND VERIFIC

Riviera-PRO QEMU environment

e <u>E</u> dit <u>V</u> iew Control <u>S</u> imulation Wav								
x • 🗎 🖪 🖬 🗶 🚺 🖪 🖕 ×		120	≪ ■ Ģ. ⊑ 219		🔔 🗘 🍄 👼			
5 🟛 🕒 🖳 🍔 🖬 🏞 😘 隆 👪				•	🗨 🗨 🔍 I 👯 🚓 I 🔬 🛛 🖄 I 🗐			
ilesystem 🔶 🗮 🗮	design_wrapper_compile.do 💌	🗙 🖟 Untitled1.awc* 💽 🗙	🗅 DpiBridge.so 💌 🗙		* 🔻 🖲 1		• • • • • • • • • • • • • • • • • • •	* 🕏 🖱
≥ = + = 2	🕂 🕂 (*• 🔍 O. 🔍 O. (*)	- 297 550 231 160 ps 👻 🛆 🗸	5 🕂 🔛 🖽 🖽 🕅	R R	N IN 🔍 👌 🐔 🐐 🐐 📶 IL 🔻 🗣 🏹 🦓 IL 🖛 🗭 🦍 Auto 🔹 1 ns	uration ^	+ +	- 字 (.*)
amples/tools/gemu/zyng demo/pl logic 🕙	Name	Hierarchy	All 🚖 Value	0	50 100 150 200 250 ms	ties ^	Name	
compile	🖶 🛥 m axi arcache	/design 1/axi cdma 0	3	3				
dataset.asdbw	m axi rready	/design_1/axi_cdma_0	0					
hdl src	m axi rvalid	/design 1/axi cdma 0	0					
riviera	🖶 🕨 maxi rdata	/design_1/axi_cdma_0	XXXXXXXX	XX	XXXXX			
work	🖽 🕨 maxi rresp	/design 1/axi cdma 0	0	x) 0			
e dataset.asdb	— ▶ m axi rlast	/design 1/axi cdma 0	х					
dataset.tmp design wrapper compile.do	■ m axi awready	/design 1/axi cdma 0	1	1				
DpiBridge.so	m axi awvalid	/design 1/axi cdma 0	0					
library.cfg	🗄 🛥 m axi awaddr	/design 1/axi cdma 0	00000000	0.0	9000			
Untitled1.do	B m axi awlen	/design_1/axi_cdma_0	00	00				
	B m axi awsize	/design_1/axi_cdma_0	0	0				
	🖶 🛥 m axi awburst	/design_1/axi_cdma_0	0					
	er • m_axi_awburst	/design_1/axi_cdma_0 /design_1/axi_cdma_0	0					
	Br = m_axi_awprot	/design 1/axi_cdma_0						
	■ m_axi_awcache	/design_1/axi_cdma_0 /design 1/axi cdma 0	3	-				
gn Manager Filesystem			0	-				
	m_axi_wvalid	/design_1/axi_cdma_0	U XXXXXXXX		90000			
ibraries 🔶 🔶 🗮		/design_1/axi_cdma_0		00	/F V0 /F			
) 🔥 🖧 🏘 📃 👻 🖓	🖶 🛥 m_axi_wstrb	/design_1/axi_cdma_0	F	0	<u>, lt , lt</u>			
ne	m_axi_wlast	/design_1/axi_cdma_0	U					
til vtl dbg (RO)	m_axi_bready	/design_1/axi_cdma_0	1					
xilinxcorelib (RO)	— ➡ m_axi_bvalid	/design_1/axi_cdma_0	Θ					
🗊 👖 xilinxcorelib_ver (RO)	⊕ ► m_axi_bresp	/design_1/axi_cdma_0	0	10				
🖳 👖 xp (RO)	L- Default cursor 219 800 us		10000000		219 800 us			
•	•			• •	0fs - 297550231160ps ► ● ● ● ● Configu	ation 🔹 🕨	4	
onsole			*	× اج	👷 Transaction Streams			* 🕏
🔥 🚓 👘 🔇 Errors 🛕 Warnings 🤇	Messages		- 9		Name Hierarchy			
	-)(start_qemu.sh; bash		×	
# KERNEL: SLP simulation initializ # KERNEL: Kernel process initializ					• *			
# Allocation: Simulator allocated		22609 kernel=6627 sdf=0)			No soundcards found. RAMDISK: gzip image found at block 0			
# VPI: ALDEC AXI BFM version 1.7.1		22005 Kernet-002, 341-0,			EXT4-fs (ram0): couldn't mount as ext3 due to feature incompatibilities			
# KERNEL: ASDB file was created in	location /home/radekn/Aldec/R	iviera-PR0-2017.06-x64/exa	amples/tools/qemu/		EXT4-fs (ram0): warning: mounting unchecked fs, running e2fsck is recommended EXT4-fs warning (device ram0): ext4_update_dynamic_rev:717: updating to rev 1 b			
zynq_demo/pl_logic/dataset.asdb					cause of new feature flag, running e2fsck is recommended			
<pre># VSIM: 176 object(s) traced.</pre>					EXT4-fs (ram0): mounted filesystem without journal. Opts: (null) VFS: Mounted root (ext4 filesystem) on device 1:0.			
<pre># VSIM: 0 object(s) traced. # WAVEFORM: 8 object(s) inserted t</pre>	a virtual array im avi rdata[A	.71			devtmpfs: mounted			
WAVEFORM: 8 object(s) inserted t					Freeing unused kernel memory: 264K (c06e6000 - c0728000)			
WAVEFORM: 8 object(s) inserted t					Starting rcS ++ Mounting filesystem			
# WAVEFORM: 8 object(s) inserted t					mount: mounting /dev/mmcblk0p1 on /mnt failed: No such file or directory			
# WAVEFORM: 4 object(s) inserted t					mount: mounting /dev/mmcblk0 on /mnt failed: No such file or directory ++ Setting up mdev			
<pre># VSIM: 0 object(s) traced.</pre>					++ Starting telnet daemon			
# KERNEL: Block Memory Generator m					++ Starting http daemon			
design_1.axi_bram_ctrl_0_bram.inst simulation which will not precisel	.native_mem_mapped_module.blk_	mem_gen_v8_3_5_inst is usi	ng a pehavioral model f	or	++ Starting ftp daemon ++ Starting ssh daemon			
simulation which will not precisel	y moder memory collision behav	101.		-	random: sshd urandom read with 0 bits of entropy available			
					rcS Complete zyng≻ ./dwa_test.bin			
onsole Messages								
onsole Messages								





Hardware and Software co-sim

<pre>in per per per per per per per per per per</pre>	ne/rad	dekn/Aldec/Riviera-PRO-2017.06-x64/examples/tools/gemu/zyng_demo/linux_app	× 🗘 🕨 🖛 🔳 🗔	🔄 📫 1 372 ms	a 🤹 😵 👼			88	✓ 66 ⁺ 66 ⁺	» ⋟ »
<pre>minute in addition to Array 1 and a first and a first and a first a first</pre>			🗖 🖩 🐁 🚊 🖣	🐼 I I 🖉 🔴 👫	🜒 🜒 🗨 i 👬 👫 i 🐔	Ē 🖓 🌐				
<pre>setime 000 AVP_SIZE block00000 setime 000 AVP_SIZE block00000 fit and DVP_SIZE block00000 fit and DVP_SIZE block00000 fit and DVP_SIZE block000000 fit and DVP_SIZE block000000000 fit and DVP_SIZE block000000000000000000000000000000000000</pre>				pper compile.do र×	🕅 Untitled1.awc* र×	DpiBridge.so TX		254		
<pre>setion BOR_NNLS_SIZE 0.03000000 setion BOR_NNLS_SIZE 0.03000000 setion BOR_NNLS_SIZE 0.03000000 int mainting reg, chart arg(1) int mainting reg, ch</pre>										
Set Marking Number (Set) Number (Set) Number (Set) 1/14ed inter (Set)				& & & & & & 13				257		
<pre>///design_last_case_of =</pre>						1360	1370 1380	258		
<pre>int main(int arg, char argv[1]) int main(int argv[1]) int main(int argv, char argv, char argv[1]) int main(int argv, char argv, char</pre>										
<pre>6 int main(int args, char* argv(1) 6 int main(int args, char* argv(1) 6 int main(int args, char* argv(1) 6 int main(int args, char* argv(1) 7 int magnet dev base, "mapped dev base, "mapped dev base, 1: 7 int main(int interve, interve, argv(1) 7 int magnet, issue 1. """" 7 interve, argv(1) 7 int magnet, issue 2. "mapped dev base 2: 7 interve, argv(1) 7 int magnet, issue 2. "mapped dev base 2: 7 interve, argv(1) 7 int magnet, issue 2. "mapped dev base 2: 7 interve, argv(1) 7 int magnet, issue 2. "mapped dev base 2: 7 interve, argv(1) 7 int magnet, issue 2. "mapped dev base 2: 7 interve, argv(1) 7 int magnet, issue 2. "mapped dev base 2: 7 interve, argv(1) 7 interve,</pre>	/	//#define BUFFER_BYTESIZE 1024								
<pre>intend() intend() intend(</pre>	4	int main(int argc_char* argv[])								
<pre>int merid; viol mapped base, mapped dev base; int merid]; viol mapped base 1, mapped dev base; int merid]; viol mapped base 1, mapped dev base; int merid 2; viol mapped base 1, mapped dev base; viol mapped base 1, mapped dev b</pre>										
<pre> fif t @up/light = mapped_mode_mess_statute int merid_1; vid # mapped_base_1 = mapped_dev_base_2; vid # mapped_base_2 = mapped_dev_base_2;</pre>		int memfd;							ATTRIBUTE X_INTERFACE_INF0	OF m_axi_wlas
<pre>int merid 1; void "mapped jases 1, "mapped der base 1; off t der base char sollt der base char sollt</pre>										
<pre>int merid 1; vicid *mapped lase 1, *mapped dev base 1; eff t dev base cfr ampped dev base 1; eff t dev base cfr ampped dev base 2; eff t dev base cfr ampped dev base</pre>		off_t_dev_base_bram = BRAM_BASE_ADDRESS;								
<pre>66 void *apped base_1, *mapped dev base_1; off_t dev_base_dn = COM_BASE_ADDRESS; 11 meref d_: void *apped base_1, *mapped dev base_2; off_t dev_base_dn = COM_BASE_ADDRESS; 11 meref d_: void *apped base_1, *mapped dev base_2; off_t dev_base_dn = COM_BASE_ADDRESS; 11 meref d_: 11 mer</pre>		int memfd 1:								"_avr_pies
<pre>off_t dv_base_dres_cdms = CDMA_BASE_ADDRESS; int dvs_base_dr = cDMA_BASE_ADDRESS; off_t dv_base_dr = cDMA_BASE_ADDRESS; off_t dv_base_dr = cDMA_BASE_ADDRESS; off_t dvs_base_dr = cDMA_BASE_ADDRESS;</pre>									UO : axi_cdma	
<pre>int exer(1_2: void mapped has 2, mapped dev bas 2; off_t dev_bas did = 00m_BASE_ADDRESS; unsigned int TimeOut =5; unsigned int fineWate; unsigned char SrcArray[10,4 1]; unsigned char SrcArray[10,24 1]; unsigned int findex;</pre>				trh		E	Ve			
<pre>c viid "mapped base 2, "mapped dev base 2; off_t dev_base df = dop. Base_ADDRESS; unsigned int TaneOut =>; unsigned int TaneOut =>; unsigned int ResetMask; unsigned int Index; intr result=0; for (Index = 0; Index < BUFFER_BYTESIZE); if (UMFFER_VTESIZE); if (Index = 0; Index < BUFFER_BYTESIZE); if (Index = 0; Index < BUFFER_BYTESIZE); if (Index = 0; Index < BUFFER_BYTESIZE); if (Index = 0; Index < BUFFER_BYTESIZE); for (Index = 0; Index < BUFFER_BYTESIZE); Index +> (</pre>		int model 2								
<pre>off_t dev_base_ddf = 00R_BASE_ADDRESS; unsigned int TimeOt_=5; unsigned int TimeOt_=5; unsigned int ResetMask; unsigned i</pre>										
<pre>d</pre>										
<pre>// unsigned int TimeOut =5; unsigned int RegValue; unsigned int RegValue; int regValue; int regValue; unsigned int RegValue; int r</pre>						0			C_M_AXI_DATA_WIDTH => 3	2,
<pre>dublighted line Materback; dublighted line Materback</pre>						6 60 00 0 00				> 16,
<pre>vmsigned int BUFFER BYTESIZE = 28; if (GUFFER BYTESIZE); sscanfargv(11, "wd", GUFFER BYTESIZE); sscanfargv(11, "wd", GUFFER BYTESIZE); sscanfargv(11, "wd", GUFFER BYTESIZE); printf('[WARNING] Passed number of bytes are wrong. Supported values BUFFER BYTESIZE); unsigned char bestArray[1024]; unsigned bestArray[1024]; unsigned bestArray[1024]; unsigned char bestArray[1024]; unsigned bestArray[1024]; bestArray[1024]; fremt(fr(int roopen /dev/mem., 0, B)KR 0, SNK); if (memfd_1 = -1); fremt(fr(int heps /dev/mem., 0, B)KR 0, SNK); if (memfd_1 = -1); fremt(fr(int int open /dev/mem., 0, B)KR 0, SNK); if (memfd_1 = -1); fremt(fr(int int open /dev/mem., 0, B)KR 0, SNK); if (memfd_1 = -1); fremt(fr(int int open /dev/mem., 0, B)KR 0</pre>						0000000 ()	X X X X X00000	277		0
<pre>9 if (argc = 2) scan(farg(1), "wd", 6&UFFER_BYTESIZE); 1 (BUFFER_BYTESIZE-10; 4 (BUFFER_BYTESIZE); 1 (BUFFER_BYTESIZE-23; 3 Unsigned char SrcArray[1024]; unsigned char DestArray[1024]; unsigned int Index; 1 If result=0; 4 (Buffer_Bytesize = 0; 10 int is 22; 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 3 7 it is 10 22 / 20 0 281 0 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 2 0 10 it is 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 10 22 / 20 0 281 0 0 1 0 1 0 10 0 2 / 20 0 10 10 10 0 22 / 20 0 281 0 0 1 0 0 1 0 10 0 2 / 20 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>						{0000} () ()				
<pre>if (BuFERE BYTESIZE-11 BUFERE BYTESIZE-1024){ print('(MARING) Passed number of bytes are wrong. Supported va BUFERE BYTESIZE-28; unsigned char SrcArray[1024]; unsigned char SrcArray[1024]; unsigned int ndex; int result=0; for (Index = 0; Index < BUFFER BYTESIZE; Index++) { STEP 1: Initialize the source buffere bytes with a pattern and clex I) Beffere ByTESIZE; Index+=) { STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes with a pattern and clex STEP 1: Initialize the source buffere bytes are source STEP 1: Initialize the source buffere bytes are source STEP 1: Initialize the source buffere bytes are source STEP 1: Initialize the source buffere bytes are source STEP 1: Initialize the source buffere bytes are source STEP 2: Nap the kernel memory location starting from 0x20000000 to i</pre>			🔲 🖽 🖽 m axi	rdata[0:7]		0	4 8 (12 (16 (20 (24)0			
<pre>22 printf(['WANING) Passed number of bytes are wrong. Supported va 23 BUFFER_BYTESIZE=28; 44 - } 45 unsigned char SecArray[1024]; 46 unsigned char SecArray[1024]; 47 unsigned char SecArray[1024]; 48 unsigned char</pre>			🖶 🖽 m axi	rdata[8:15]		θ (1)	5 9 13 17 21 25 0			
<pre>ButFER_BYTESIZE=28; butferg_Bytesize=28; but insigned char_SrcArray[1024]; unsigned char_Destarray[1024]; unsigned int result=0; but int result=0; bu</pre>			n 🖬 🖬 m axi	rdata[16:23]		θ (2)	6 10 14 18 22 26 0			20
33 } 34 } 35 unsigned char SrcArray[1024]; unsigned char DestArray[1024]; unsigned in thex; int result=0; 325 C_DCYTMER_ESSOLUTION >> 2 36 unsigned char SrcArray[1024]; unsigned in thex; int result=0; 326 C_AMILY >> "zynq" 37 unsigned char SrcArray[1024]; unsigned in thex; int result=0; 326 C_AMILY >> "zynq" 38 int result=0; int result=0; int result=0; 326 C_AMILY >> "zynq" 39 //// int result=0;			🖽 🖽 m_axi	_rdata[24:31]		0 (3)	7 11 15 19 23 27 0			
<pre>start_genush; bash v p (unsigned char SrcArray[1024]; unsigned char DestArray[1024]; unsigned int result=0; v=</pre>			- Default cursor	1 372 ms			1 372 ms		C_DLYTMR_RESOLUTION =>	
<pre>Bit content of the source of the source buffer bytes with a pattern and clear bit coation bit coa</pre>			4			Daberteateateateat	1200240500425=	286	C FAMILY => "zynq"	
<pre>ass =</pre>						start_qemu.sh; bash			× P (
<pre>int result=0; int result=</pre>				zung> ./dma test.bin					i aclk => m axi acl	k.
90 B /*====================================				[i] Mapping PL logic B	RAM memory.					
<pre>SIP 1 initialize the source buffer bytes with a pattern and cle location for (Index = 0; Index < BUFFER_BYTESIZE; Index++) {</pre>	Ξ			BRAM memory ma	pped to user space.					
<pre>33 for (Index = 0; Index < BUFFER_BYTESIZE; Index++) 35 { 36 { 37 DestArray[Index] = Index; 38 - } 38 - } 39 /*===================================</pre>				writing data 2	8 to BRAM.					
4 for (Index = 0; Index < BUFFER_BYTESIZE; Index++)	L		=0)	[I] Mapping AXI CDMA t	sned, newory unwapped, o user space					
95 = { srcArray[Index] = Index; DestArray[Index] = 0; srcArray[Index] = 0; srcArray[Inde		<pre>for (Index = 0; Index < BUFFER BYTESIZE; Index++)</pre>	l/examples/tool	/dev/mem opene	:d.					
37 DestArray[Index] = 0; 38 } 39 /*=	Ξ		Wexamptes/toot.	enabling inter	rupt.					
<pre>set escurce addres to BRMH = 81000000, set destination address to BRMH = 81000000, set destination address to BRMH = 8000000, set destination address to BRMH = 8000000, if the four paped at address to BRMH = 8000000, if the four paped at address to BRMH = 8000000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 8000000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four paped at address to BRMH = 800000, if the four pa</pre>				checking for t	he Bus Idle.				i lite wvalid => s	axı lite wval
<pre>set destination address to hr_LATK = doublood. set destination address to hr_LATK = doublood. wait for Diff transfer Coopleted printf("[i] Mapping PL logic BRAM memory.\n"); memfd_l = open("/dev/mem", 0_RDWR 0_SYNC); if (memfd_l == -1) f = printf("Can't open /dev/mem.\n"); exit(0); </pre>	L	<pre>vestarray[index] = 0; }</pre>		set source add	res to BRAM = 81000000.				5.0 150 8850 1	✓ Unix INS
00 STEP 2: Map the kernel memory location starting from 0x20000000 to 10 printf("[i] Mapping PL logic BRAM memory.\n"); 10 printf("[i] Mapping PL logic BRAM memory.\n"); 10 memfd_1 = open("/dev/mem", 0_RDWR 0_SYNC); 11 Memory hore data read from 0x20000000 to 15 { 16 printf("can't open /dev/mem.\n"); 17 exit(0);	Ē	, /*		set destinatio	n address to HP_SLAVE = 80010	0000.			50 150-6659-1	* Unix INS
<pre>printf("(i) Mapping PL logic BRAM memory.\n"); memfdl = open("/dev/mem", 0_RDWR 0_SYNC); if (memfd_l = -1) 5 = { printf("Can't open /dev/mem.\n"); printf("Can't open /dev/mem.\n"); cxit(0); cxit(0);</pre>				Transfer Completed						
<pre>// Inf (if happing to togic block means y, (if y, memfd_1 = open ("/dev/mem", 0_RDWR 0_SYNC); if (memfd_1 == -1) 5</pre>	E F					pace.				
04 if (memfd_1 == -1) 05 E 06 printf("Can't open /dev/mem.\n"); 07 exit(0);				/dev/mem opene	:d.					
05 0				Memory mapped	at address 0xa6e33000.					
96 print("Can't open /dev/mem.\n"); 97 97 exit(0);	-	{	using a behavi	The data pattern:						
The data read from HDL:			, asting a beliav.	0123456789101112131415	161718192021222324252627					
		exit(0);		The data read from HDL	:					
<pre>99 printf("\t/dev/mem opened. \n");</pre>	F	<pre>} printf("\t/dov/mom_opened_\n");</pre>		0123456789101112131415	161718192021222324252627					
by print(((t/dev/mem opened. (h');		princit (c/uev/mem openeu. (n /;			rsstul, 20					



DESIGN AND VERIFIC

Transaction based debug

						Untitled1.awc*						
e <u>E</u> dit W <u>a</u> veform												(
+ ÷- K 🔍 🔍 🔍	🔍 🔍 🛯 351 110 666 98	7 ps 💌 🛆 🛆 🧏 🐺	II II II 🕅 🕅 🕅	M IIP I Q 🕞	n 🖓 🦂 🐜 %	л - 💠 🐐 🐝	Auto 👻 🛙 n	S	÷			
ne	Hierarchy All 🌲	Value	1336		1338	1340	1342		1344	1346	1348	1350
	/design_1/axi_cdma_0	0	0									
	/design_1/axi_cdma_0	0	0									
■ m_axi_awprot	/design_1/axi_cdma_0	0	0									
	/design_1/axi_cdma_0	3	3								•	
	/design_1/axi_cdma_0	1										
	/design_1/axi_cdma_0	1	03020100		07060504	08040908	OFOEODO	c	V13121110	17161514	V1B1A1918	00000000
	/design_1/axi_cdma_0 /design 1/axi cdma 0	0F0E0D0C	03020100		/07060504	(0D0A0900	JOFOEODO		/13121110		VIDTATATO	00000000
• m_axi_wstrb • m axi wlast	/design_1/axi_cdma_0 /design 1/axi cdma 0	г 0	F									<u>,0</u>
	/design_1/axi_cdma_0 /design_1/axi_cdma_0	1										
m_axi_bready m_axi_bready	/design_1/axi_cdma_0 /design_1/axi_cdma_0	0										
m_axi_bvattu m_axi_bresp	/design 1/axi_cdma_0	0	0									
'		00000000	00000000									
▶ m axi rdata		13121110	03020100 070605	04	08040908	0F0E0D0C	1312111	0	17161514	181A1918	0000000	
■ Virtual Record 1		{16 17 18 19}	{0 1 2 3} {4 5 6		{8 9 10 11}	{12 13 14 15}	(16 17		{20 21 22 23}	{24 25 26 27}	(0 0 0)	
			✓ Count (2) ✓			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			
				Relations V Auto	Theight							
	/design_1/slave_0/Tran⇒							W_Stream				
Sender name		design_1.slave_0						gn_1.slave_0		⊟ design_1.slave_θ		design_1.
Sender WID Sender WDATA		zZ0 0706050403020100					zZ0	050403020100		zZ0 0f0e0d0c0b0a0908		relations
Sender WDATA Sender WSTRB		0706050403020100 ff					6706 ff	050403020100		ff		relations
Sender WLAST		0					0			0		
Sender WVALID		1	1				1			1		
Sender WREADY		î					1			1		
Signal		Value					Valu			Value		
relations in(0)		Vatac						ions in(0)		relations in(0)		
relations out(0)								Lons out(0)		relations out(0)		
											-	
			2						Sender name : design_1 Sender WID : zZ0	.slave_0		
									Sender WDATA : 070605	0403020100		
									Sender WSTRB : ff			
			🚔 🗸 Count (1) 🖌	Relations 🖌 Auto	o height				Sender WLAST : 0 Sender WVALID : 1			
R_Stream	/design_1/master_0/Træ							R_Stream	Sender WREADY : 1			
									Signal : Value			
									Stream: sim:/design_1/si Substream: 1	ave_0/TransRecorder/W_Strear	n	
									Begin time: 1342ms			
									End time: 1344ms			
			1						1			
									1 Value			
									relations in(0)			
									relations out(0)			
									returions out(o)			
			🚔 🖌 Count (1) 🖌	Relations 🖌 Auto	o height							
AR_Stream	/design_1/master_0/Træ							AR_Stream				
											⊡ design_1.master_θ	
											000	
											7e200004	
											0	
Default cursor 13	42 650 us						1 34	2 650 us			1 2	1
			▶ ◀ 133502933301	3ps - 1351110666	5987ps							• • •
				562 1991110000								

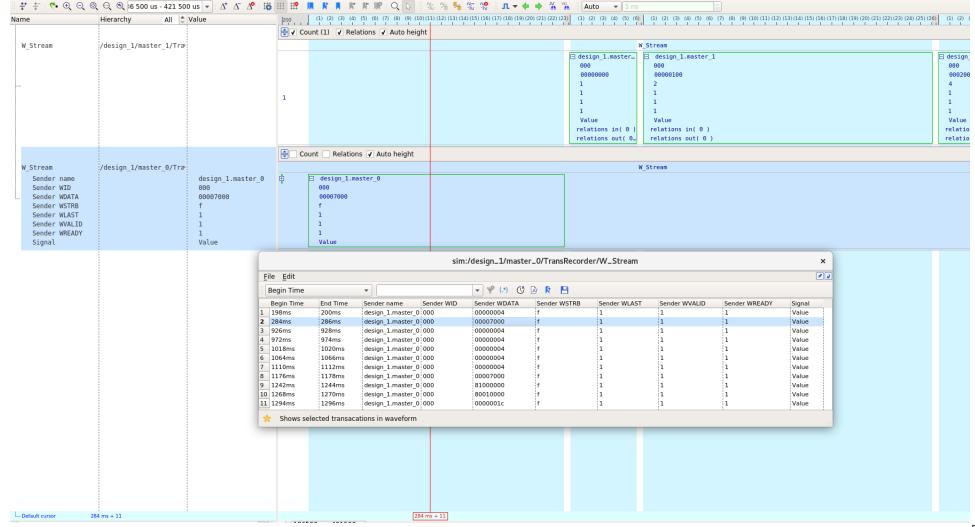


DESIGN AND VERIFICATION

CONFERENCE AND EXHIBITION

Transaction logs

<u>File Edit Waveform</u>

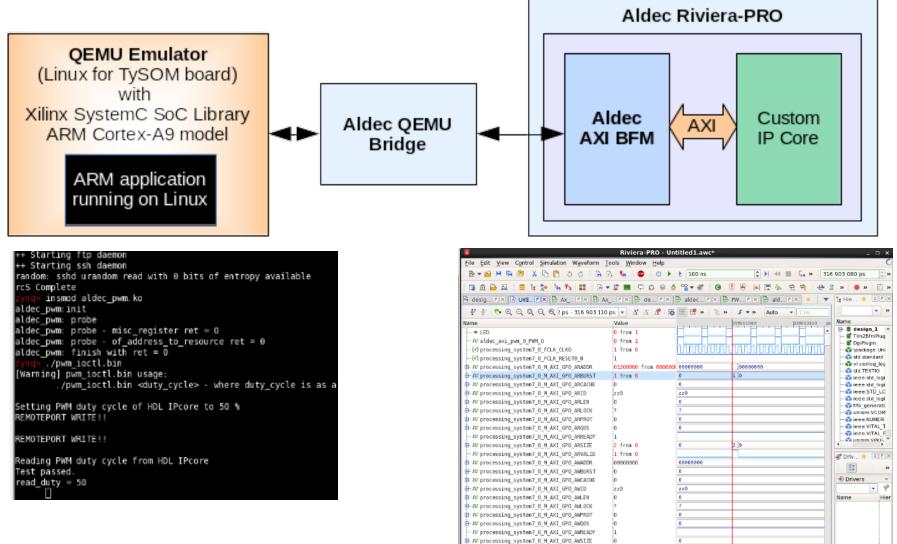




20



Riviera-PRO and QEMU – project flow





- DEprocessing system7 0 M AXT GPO AWAI TO

2017

DESIGN AND VERIFICATION

CONFERENCE AND EXHIBITION

EUROPE

	TySOM-1	TySOM-2	TySOM-2A
Boards Specs			
SoC	XC7Z030	XC7Z045/100	XC7Z030
DDR3	FBG484 512MB	FFG900 1GB	2FFG676l 1GB
SPI Flash	16MB	16MB	16MB
uSD/SD/EMMC	uSD	uSD	uSD
I2C EEPROM	64Kb	64Kb	64Kb
USB 2.0	2	4	4
USB 3.0	2	-	-
UART/USB → UART	2 UART	1 USB \rightarrow UART	1 USB \rightarrow UART
JTAG PL Header	1	1	1
1 Gb Ethernet	V	V	v
Ethernet IEEE1588	-	-	v
Audio IN/OUT	V	-	-
HDMI/VGA	HDMI	HDMI	HDMI
Camera Con	V	-	-
mPCIe	V	-	
FMC	-	× 2	× 1
Pmod	2 × 6	-	-
GPIO	43 (*)	-	14(**)
GTX	1 on MMCX con	16 on FMC con	4 on FMC con
ADC	v	V	v
Wi-Fi/Bluetooth	-	-	v
Sensors	Temp. Accelerometer	Temp. Accelerometer	Temp. Accelerometer
RTC	v	V	v
TouchPannel/LCD Interface	T.P & LCD	-	-
User DIP switch	8	8	8
User LED Available on mother boards	8	4	4

Available on mother boards / expansion carrier board. Available on YADC blooder.

**Available on XADC Header.



TySOM – Zynq Based EDK

\backslash	FMC – IoT	FMC – Vision	FMC – ADAS	FMC – INTF
Boards Specs			000	
	64 Kb EEPROM, uSD,	64 Kb EEPROM	64 Kb EEPROM	4 Gb NAND Flash, 2×
Memory	USER SATA			32 Mb SPI Flash
Michiory				Memory, 64 Kb
				EEPROM, uSD
	FMC, 1× Ethernet	FMC, 1× HDMI IN, 1×	FMC, 5× FPD-Link III	FMC, 1× Ethernet,
	IEEE1588, Wi-Fi,	HDMI OUT, 1×	(LVDS interface), 1×	PCI-Express
	Bluetooth 4.1,	Display Port IN, 1×	URM37_V4.0	connector, QSFP+
	ZigBee, 6LoWPAN,	Display Port OUT,	Ultrasonic sensor	connector, 1× USB
Communication	RF4CE, SP100,	MIPI DSI, MIPI CSI,	connector, 1×	2.0, 1× UART to USB
interfaces	WirelessHART,	Audio Codec, 2× USB	LIDAR-Lite_V2	
inter inces	Sigfox, WMBus,	3.0	Connector	
	mPCle, USB 2.0,			
	Micro Sim card, RS-			
	485, 2×6 Digilent			
	Pmod			
	2× user clock, 1×	2× Oscillators, 2×	Buzzer, 2× Oscillator	3× Oscillator, 8× User
	clock for SATA,	Clock for		DIP switch, 8× User
Miscellaneous	Dedicated clock to	transceivers		LED, GPIO (20 IOs,
	PCIE			2×20 gold pins
				connector)



Summary

- SoC FPGA Verification Platform
- Processor modeling based on QEMU emulator (instead of simplified BFM model)
- Hardware and Software debugging
- Fast RTL IP Core Verification and bugs fixing
- Driver and Firmware Verification
- Built-in Transaction Recorder
- Support for many platforms and processors
- Support for different Linux versions
 - Yocto project





Resources

Trainingwww.aldec.com/trainingSupportwww.aldec.com/supportBlogwww.aldec.com/company/blog

Questions



