2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

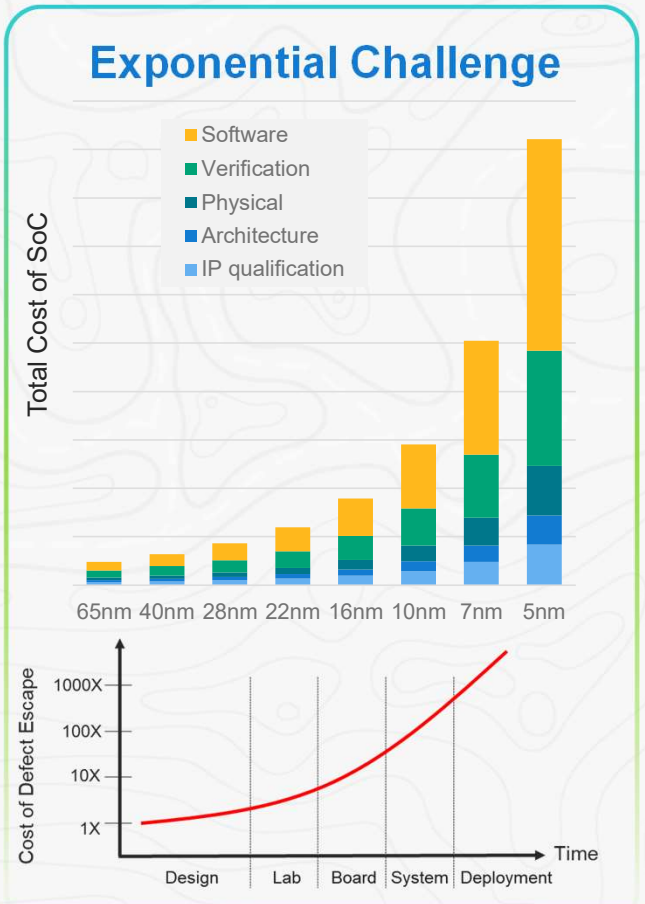
Virtual Platforms to Shift-Left Software Development and System Verification

Ross Dickson, Pankaj Kakkar
Cadence Design Systems, Inc.

cadence®

accellera SYSTEMS INITIATIVE

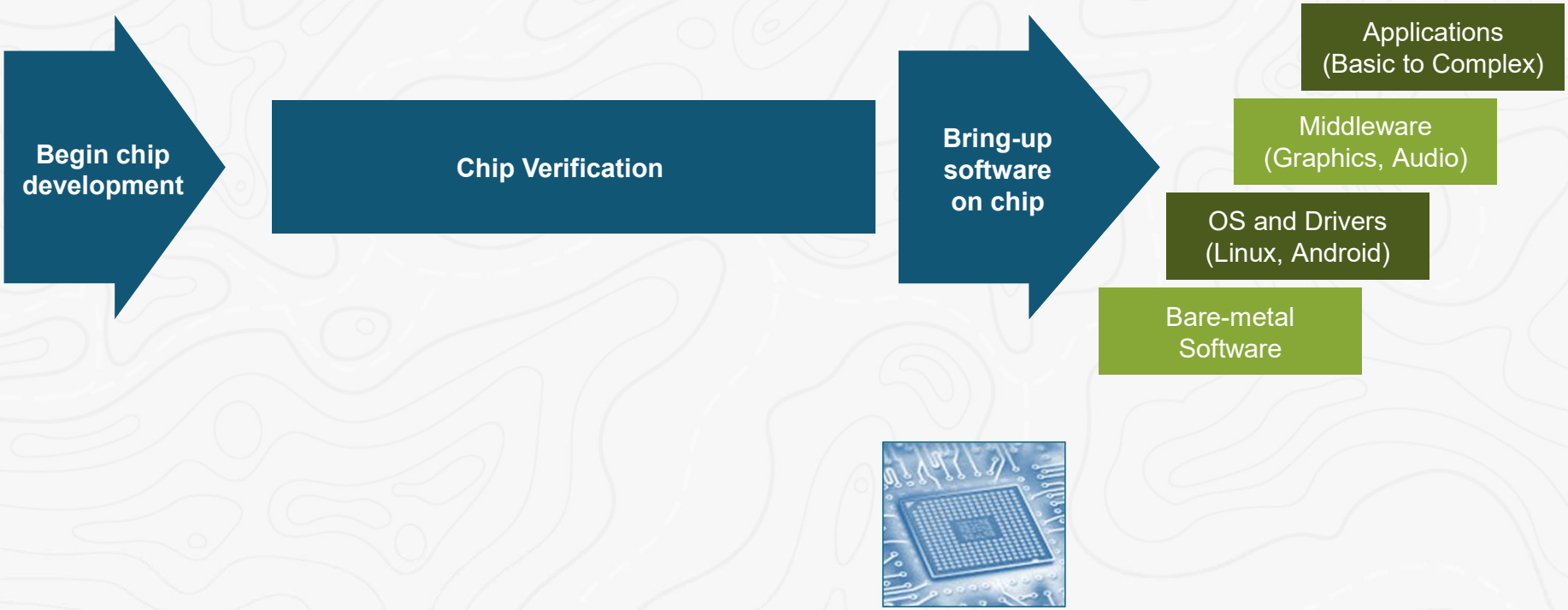
The Problem We All Know



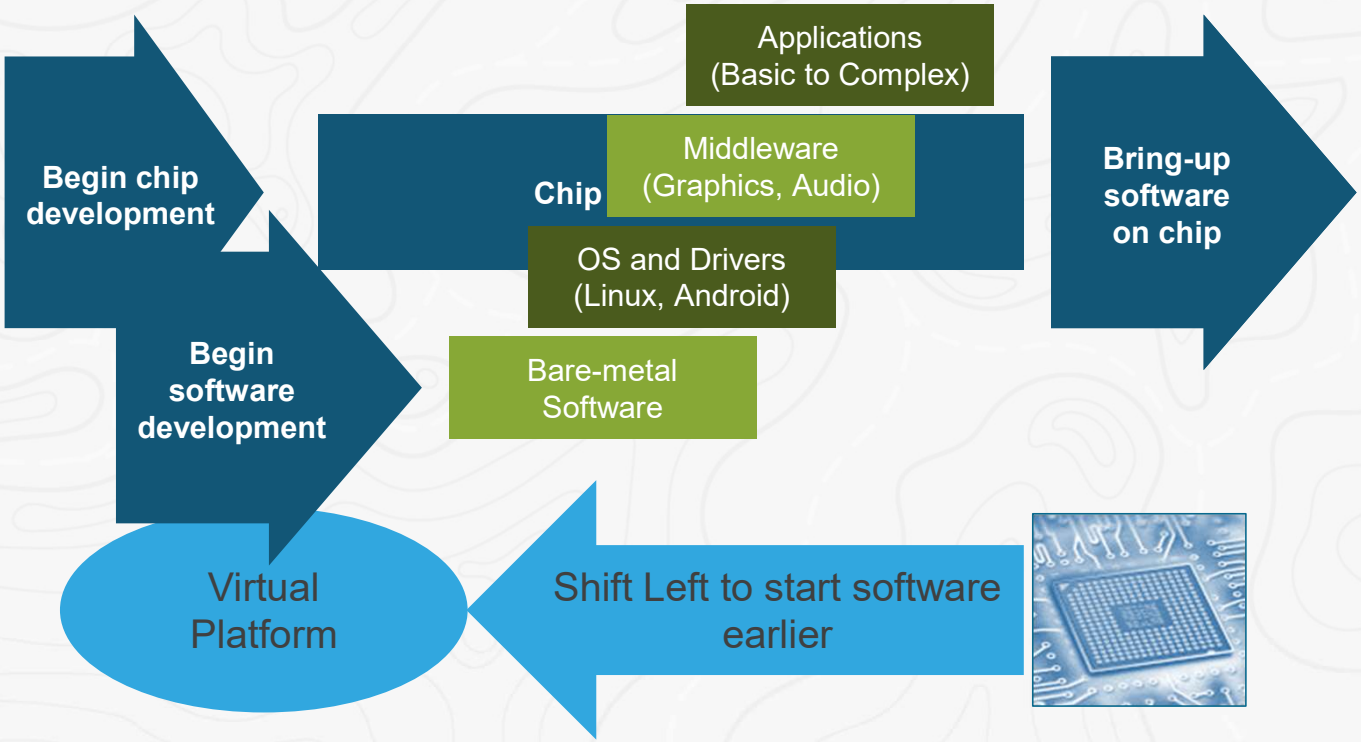
ROI Mindset: bugs found per $ per day

VERIFICATION THROUGHPUT

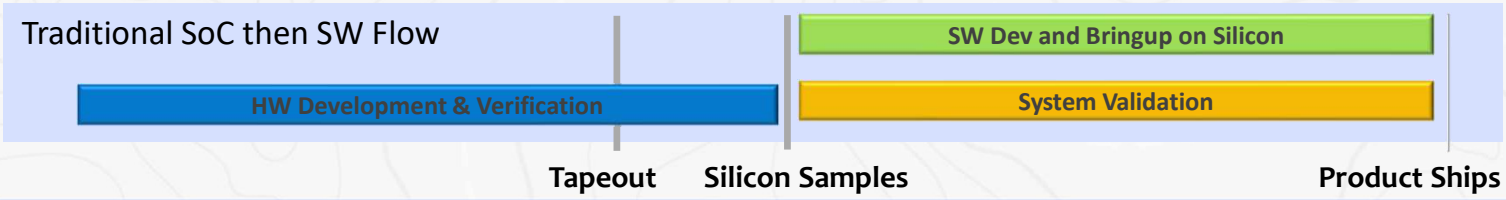
Shift-Left: the Solution We All Talk About



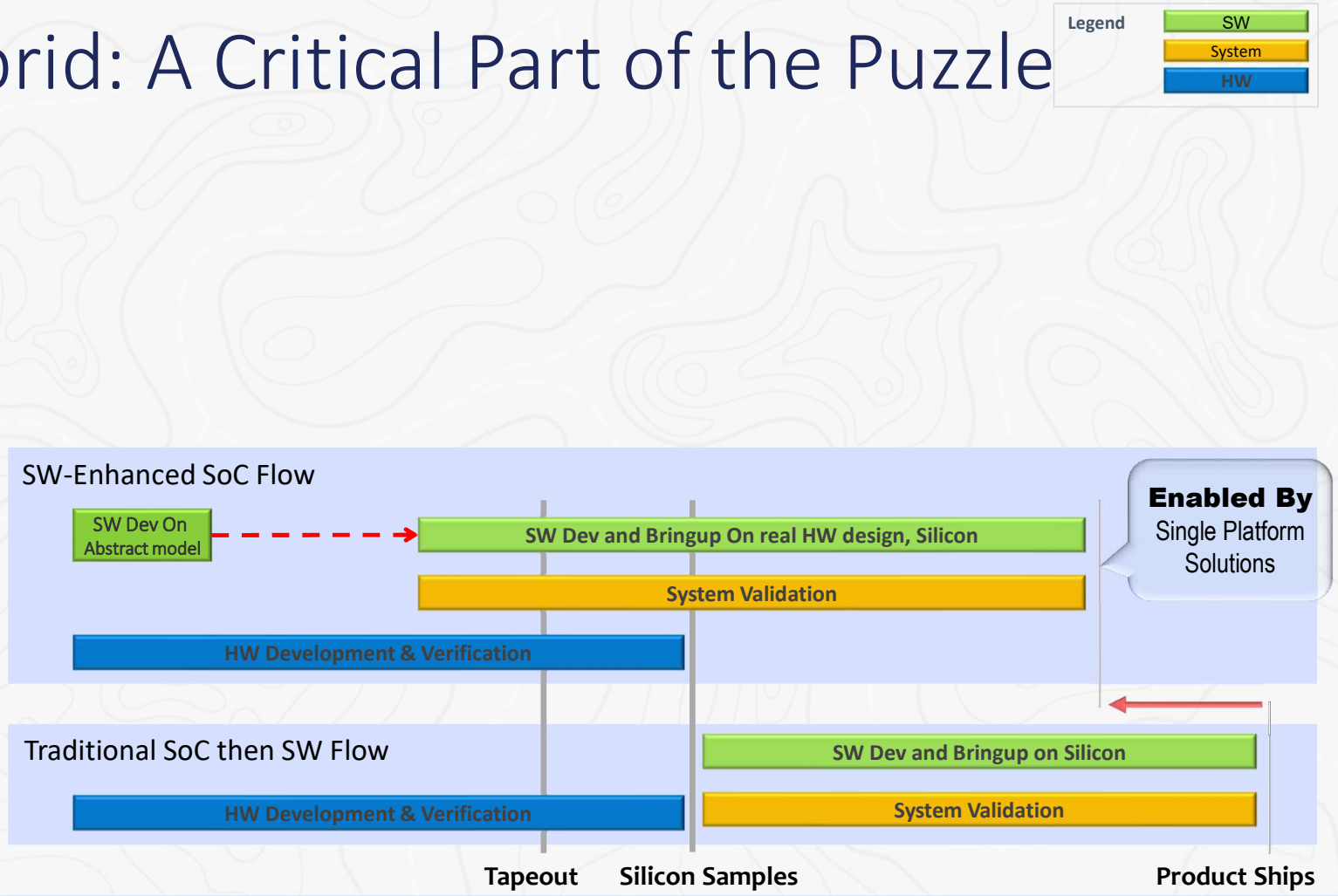
Shift-Left: the Solution We All Talk About



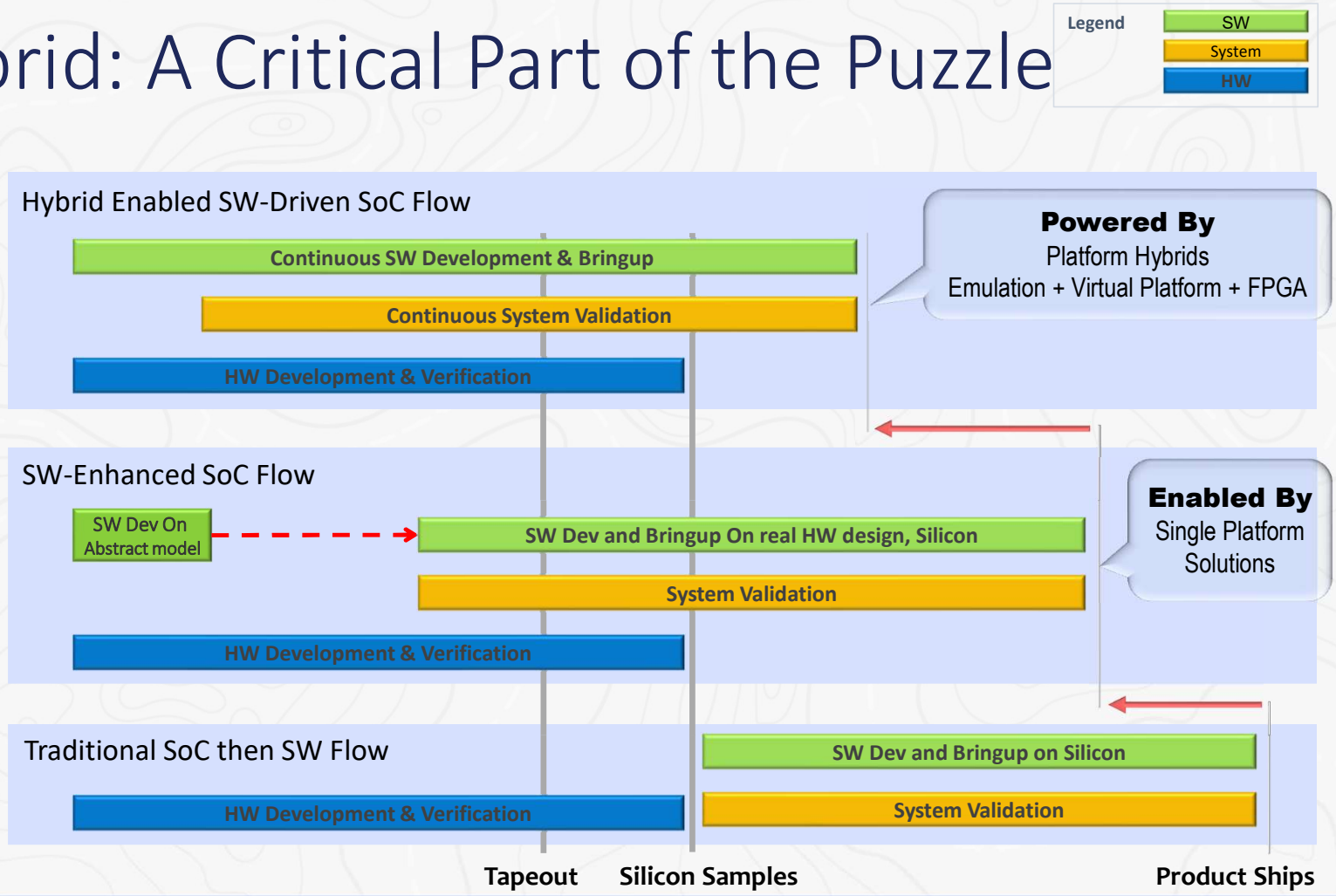
Hybrid: A Critical Part of the Puzzle



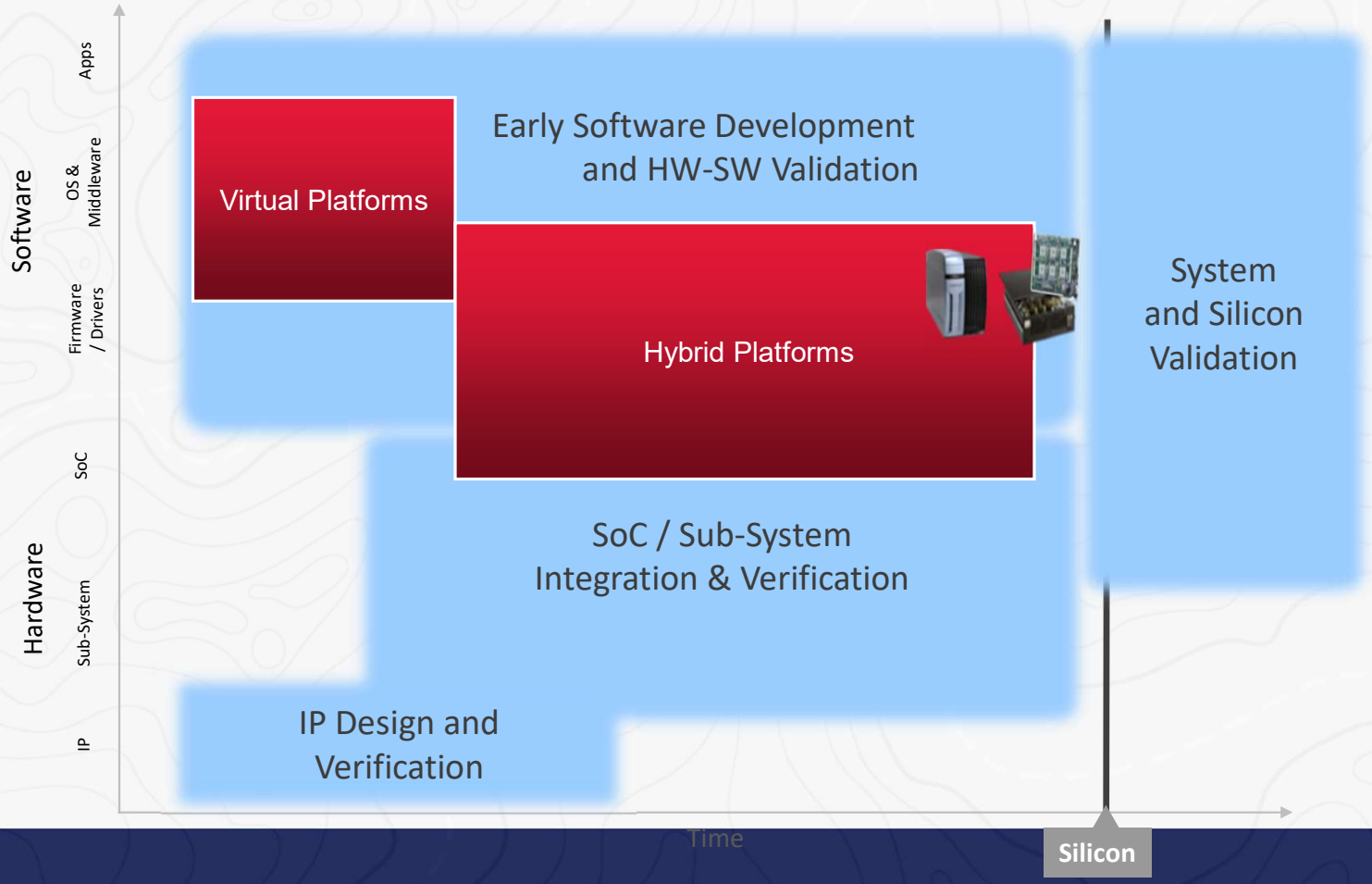
Hybrid: A Critical Part of the Puzzle



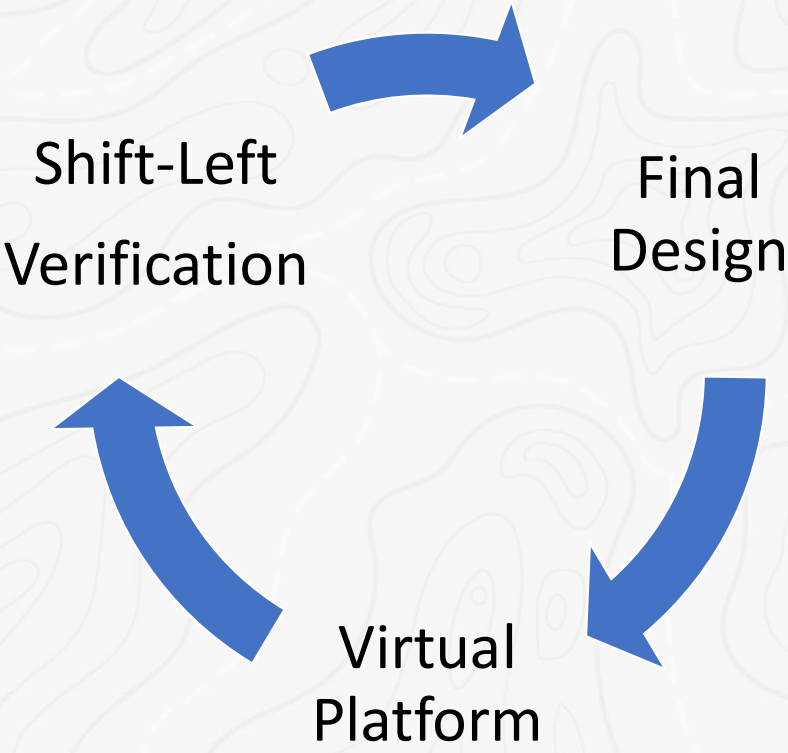
Hybrid: A Critical Part of the Puzzle



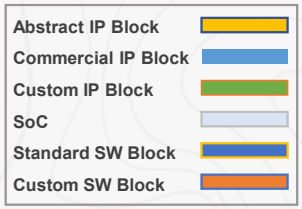
Where is Hybrid Applicable?

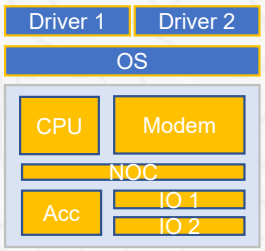


What About the Cyclic Dependency?

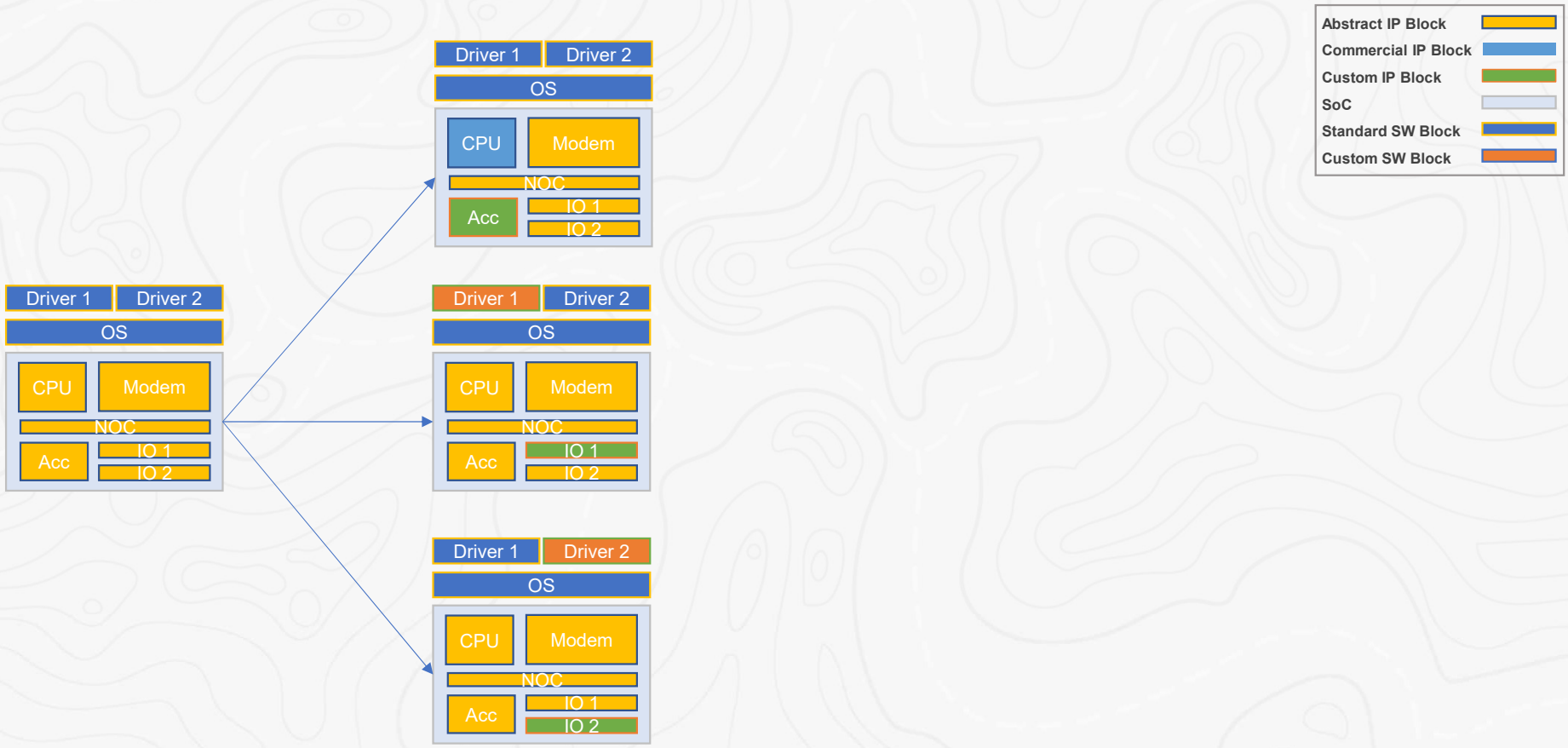


Successively Refined Abstract Platforms

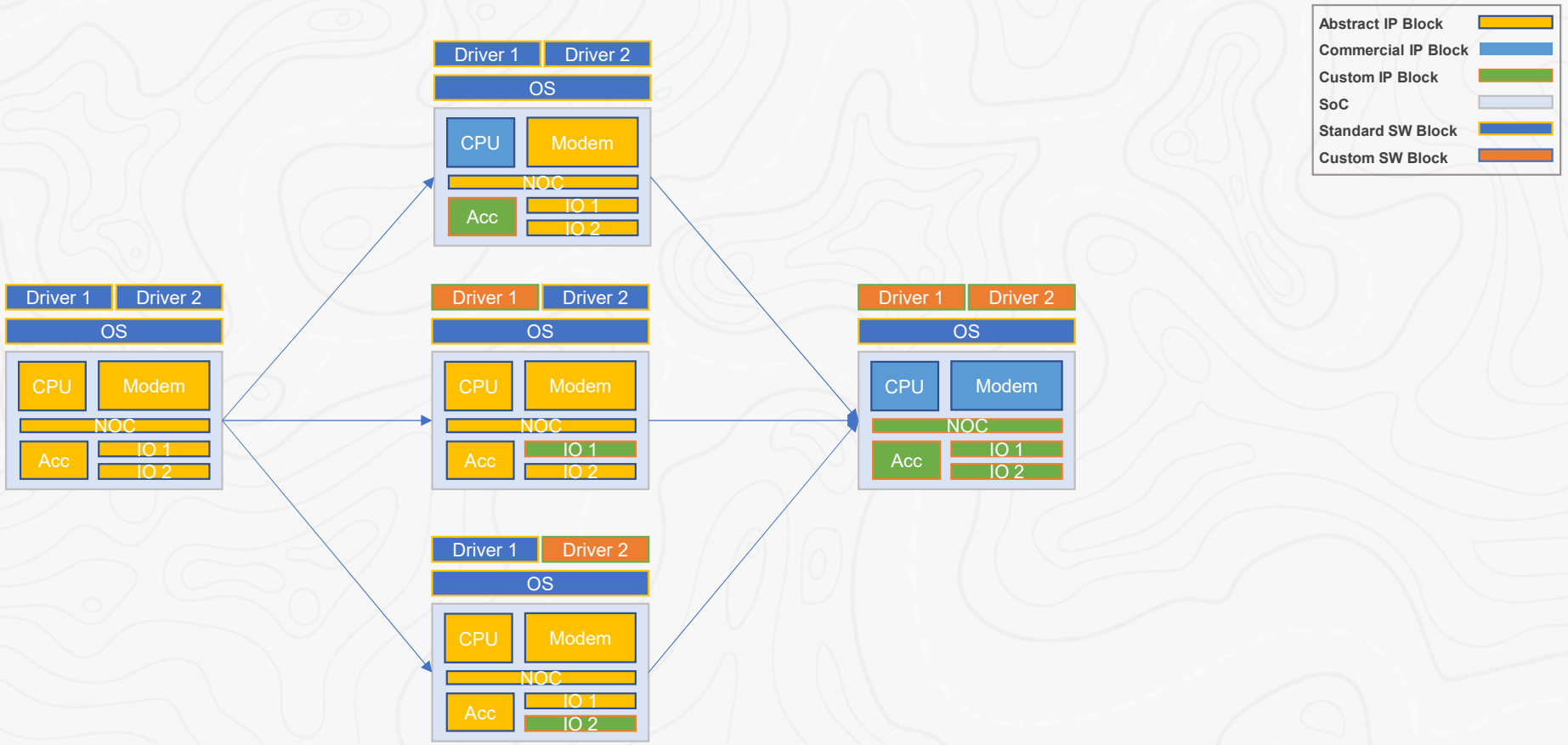




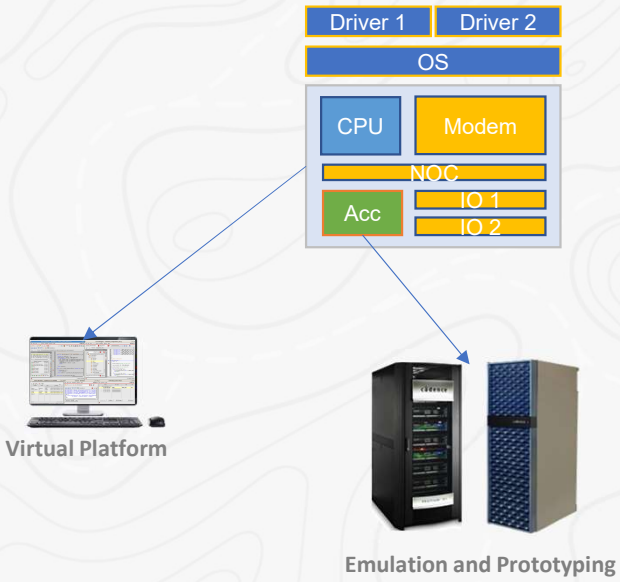
Successively Refined Abstract Platforms

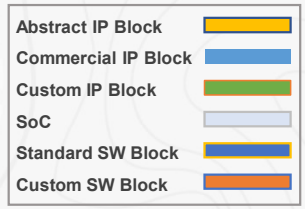


Successively Refined Abstract Platforms



Extend Accuracy with Hybrid Platforms





- Extend with Emulation and FPGA Prototyping to run with the most accurate representation of the design ASAP using platforms running 100's of MHz
- Typical Use-cases
  - Bring-up and test Linux and drivers
  - Bring-up and test Android
  - Targeted benchmarks and performance optimization
  - Critical application bring-up

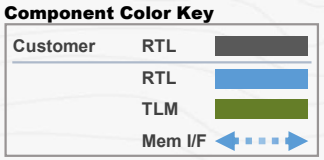
Example Hybrid System

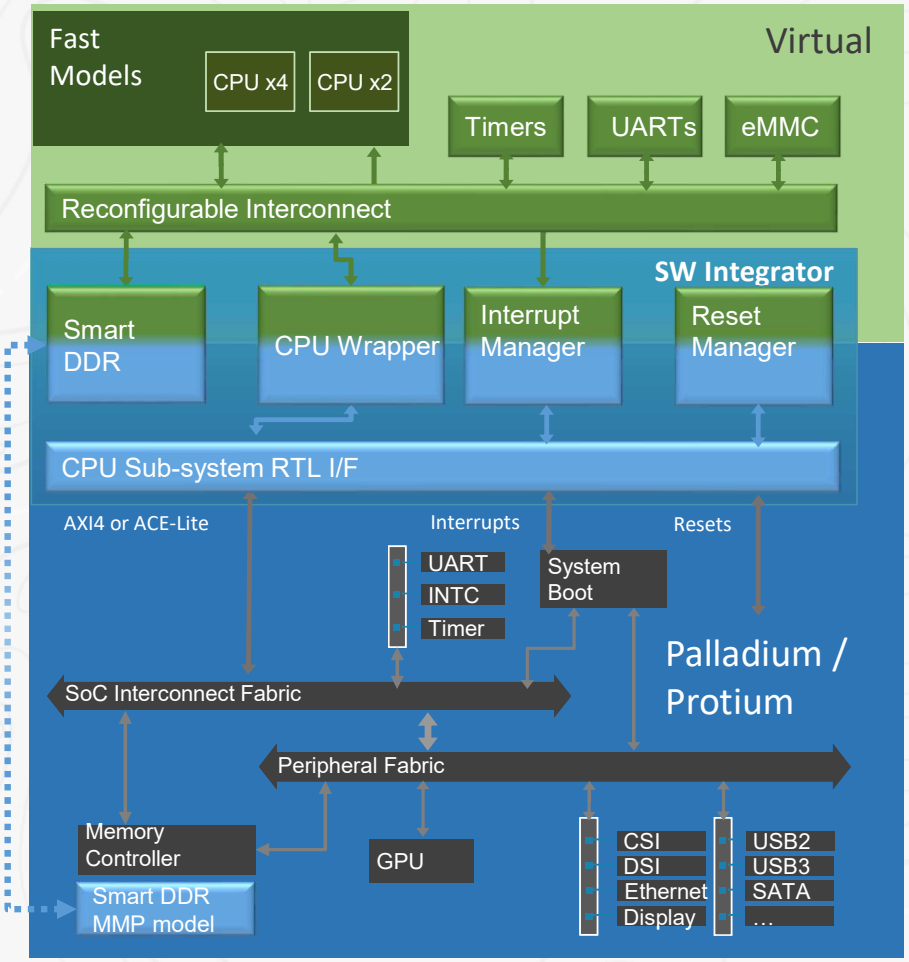
Execute SW at 100MHz
With standard or custom processor models

Plug and Play Integration with RTL
SoC-specific transactors and RTL I/F

Validate SoC + OS at 5-10 MHz
High-performance memory coherency

Reduce SoC Debug Effort
System Messages
HW / SW Debuggers

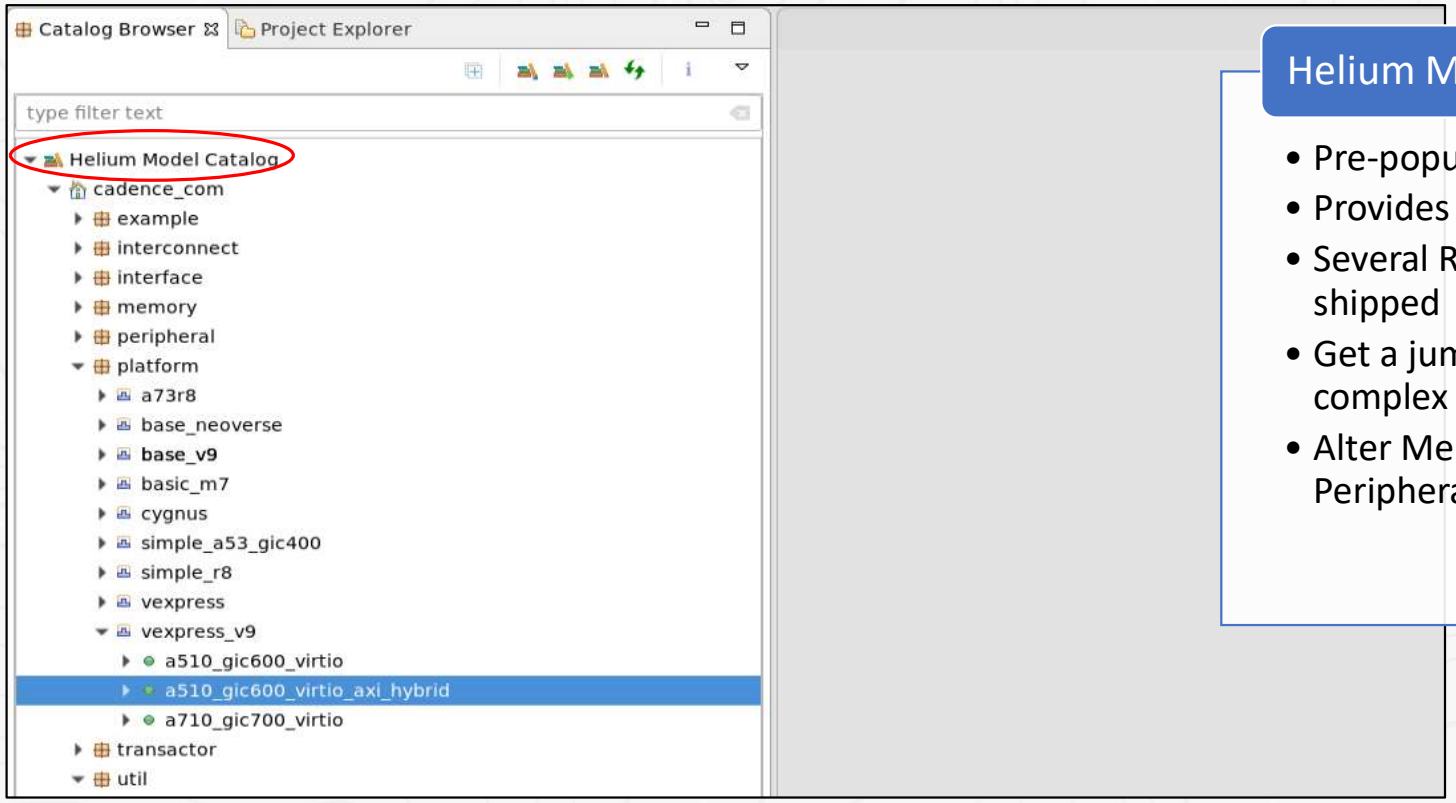




Typical Hybrid Platform Creation and Software Bring-up Flow

- Create New Hybrid Platform
  - Start with Example hybrid platform
  - Save-As flow for new hybrid platform
  - Add/Remove virtual models (as needed)
  - Configure parameters and bind ports
  - Rewire interrupt and memory map
  - Add/Remove RTL stubs (as needed)
  - Configure Hybrid Adaptors for connecting to new RTL design
- RTL changes to connect Helium Hybrid Adaptors
  - CPU Architecture Example
  - CPU RTL wrapper details and configuration options
  - GIC wrapper (GIC600)
  - Smart Memory (LPDDR5)
- Build & Run Linux on Hybrid Platform and enable SW debug

Getting started with Example: A510 Hybrid Platform



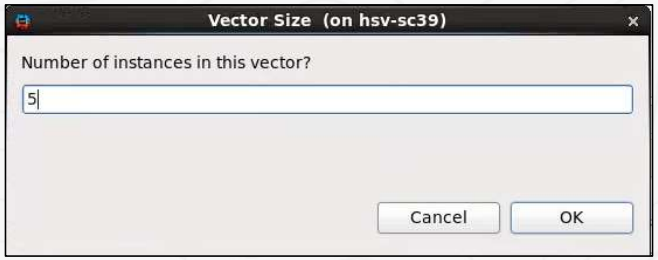
Helium Model Catalog

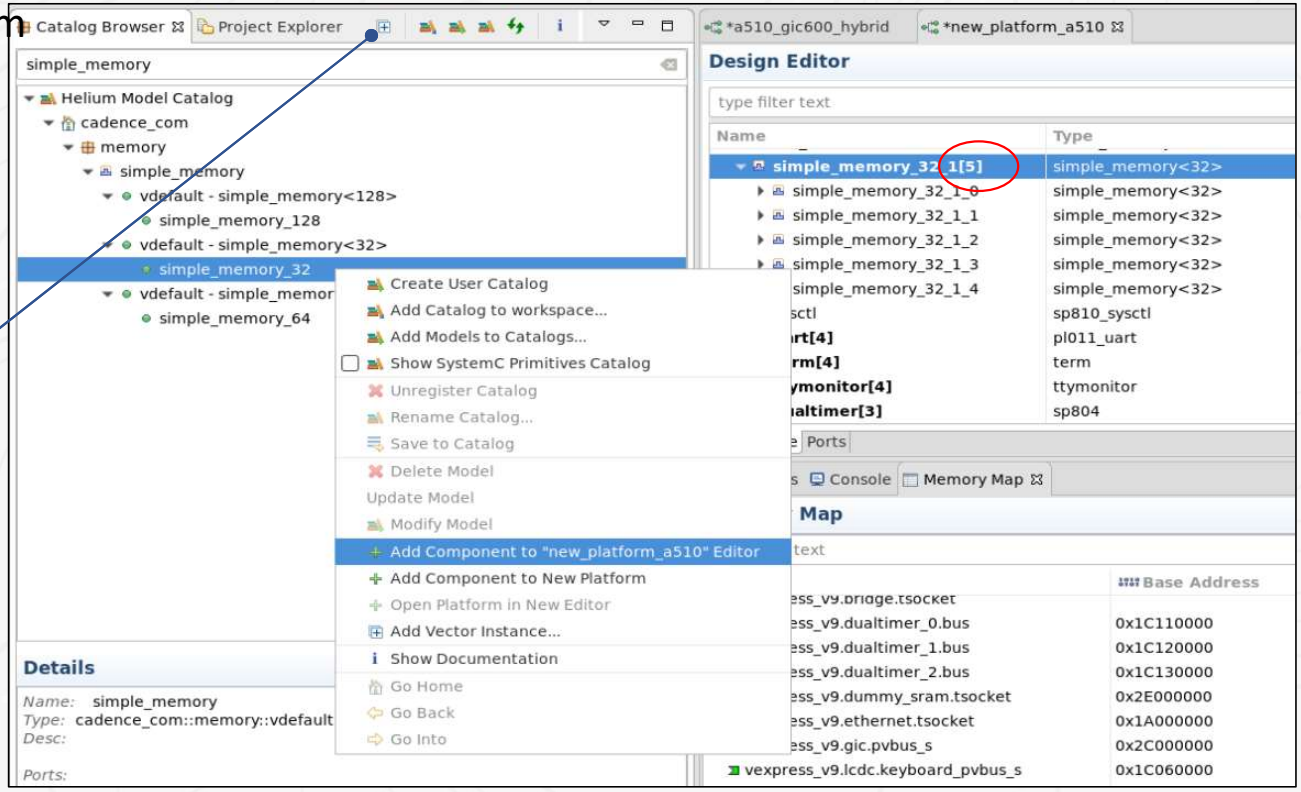
- Pre-populated in GUI
- Provides several Base Models
- Several Reference Platforms examples shipped
- Get a jumpstart and quickly build complex design
- Alter Memory Map, add/remove Peripheral as needed

Add/Remove virtual models (1/2)

- Adding model Instances to a platform in Helium with Search/Filter
- Simply find the model you want and double-click.

Use “Add Vector Instance” to create an sc_vector of instances





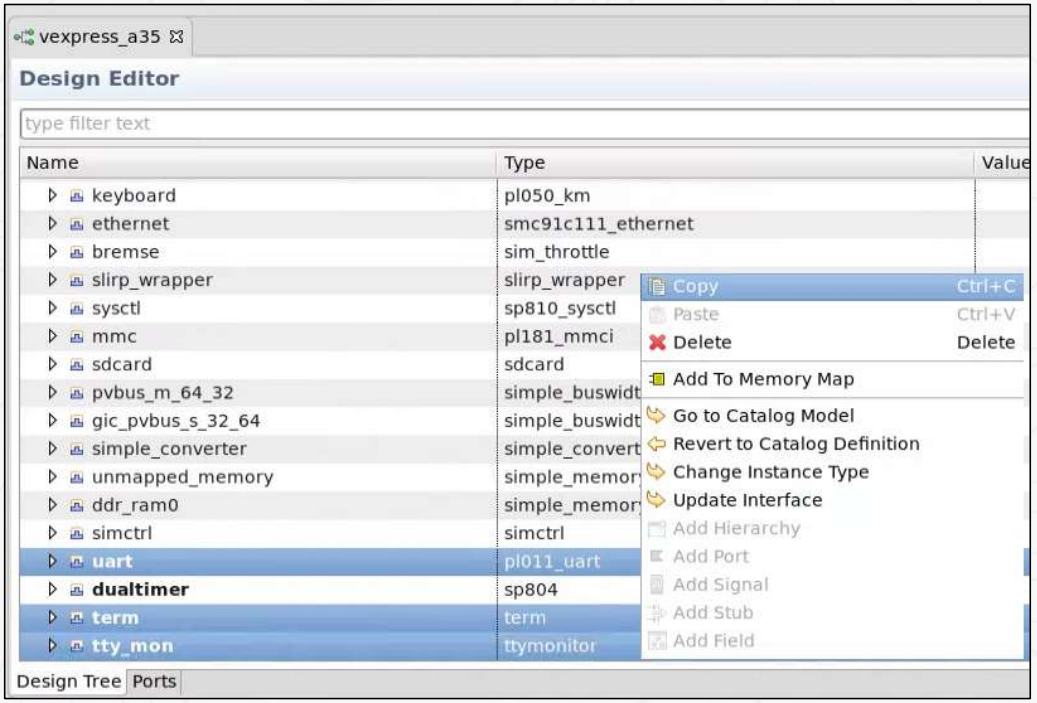
Add/Remove virtual models (2/2)

Copy and paste with connections also creates new instances

- When you want to save work by re-using groups of connected instances from other platforms, simply copy a group of connected instances and paste them into your platform
- No need to redo connections between components and set parameter values – they get copied with the group
- Make sure to make any **required port connections** from ports in the group to **external instances**

Select the uart, term, tty_mon instances in the design editor, right-click and choose Copy or use <Ctrl-C>

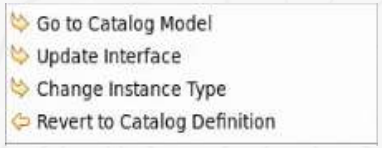
To create a new instance of the selected components, choose a platform in an open editor, select the appropriate scope, right-click and choose Paste or use <Ctrl-V>



Parameters Configurations

- Visualize platform instance parameters and easily configure their values

Use Update Interface to reconfigure instance when you change parameters that affect model interface



Platform Assembly Design Editor

Design Editor

type filter text

Name	Type	Value	Details
ARM_Fast_Models_Global_Parameters	ARMPlatform		
Parameters (12)			
tlm_global_quantum	double	0.000100	The system global simulation quantum time in seconds. Set either this or quantum_seconds param
scx_set_min_sync_latency	double	0.00001	The minimum synchronization latency. Measured in seconds
quantum_seconds	double	0.001	The system global simulation quantum time in seconds. Set either this or tlm_global_quantum para
scx_start_cadi_server_start	bool	false	Set to true to start a CADI server
scx_start_cadi_server_run	bool	false	Set to true to run the simulation immediately after CADI server has been started
scx_start_cadi_server_debug	bool	false	Set to true to enable debugging through a plug-in
scx_load_application_all	std::string		The application to load
scx_load_plugin	std::string		The file of the plug-in to load
scx_print_statistics	bool	true	Set to true to enable printing of simulation statistics
scx_enable_cadi_log	bool	false	Set to true to log CADI calls
scx_parse_and_configure	bool	false	Parses command-line options and configures the simulation accordingly
tarmac_trace	bool	false	Fast Models Tarmac Trace

Global parameters common to all ARM Fast Model instances

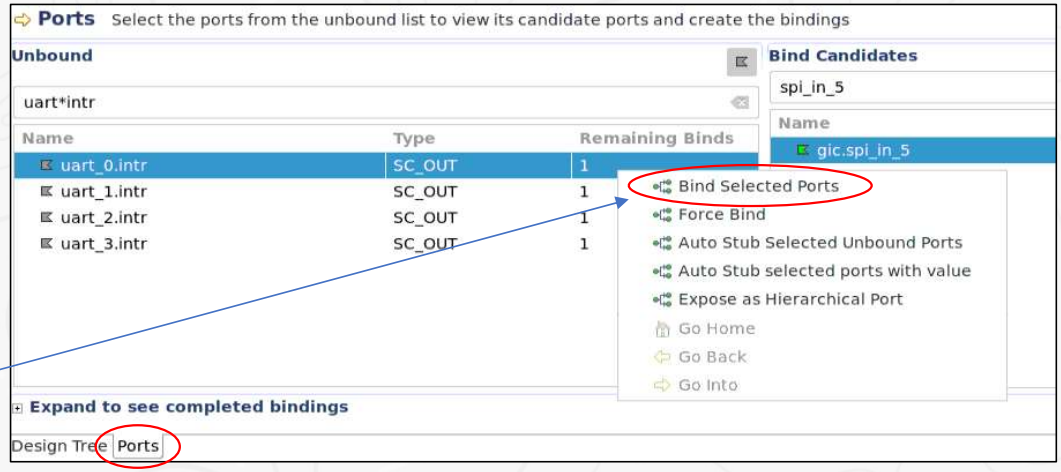
View and edit individual model instance parameters

sram	simple_memory<32>		
Ports (2)			
Parameters (10)			
verbosity	std::string		Per-instance verbosity, always overrides global verbosity. Range: {0,100,200,300,400,500} or {SC_NON
size	unsigned long long	0x2000000	Set size of simple memory in bytes. Range: {0x1,...,0x100000*,0x100001,...,2^63}
enable_dmi	bool	true	Set to true to enable DMI. Range: {true*,false}

Handling Port Bindings

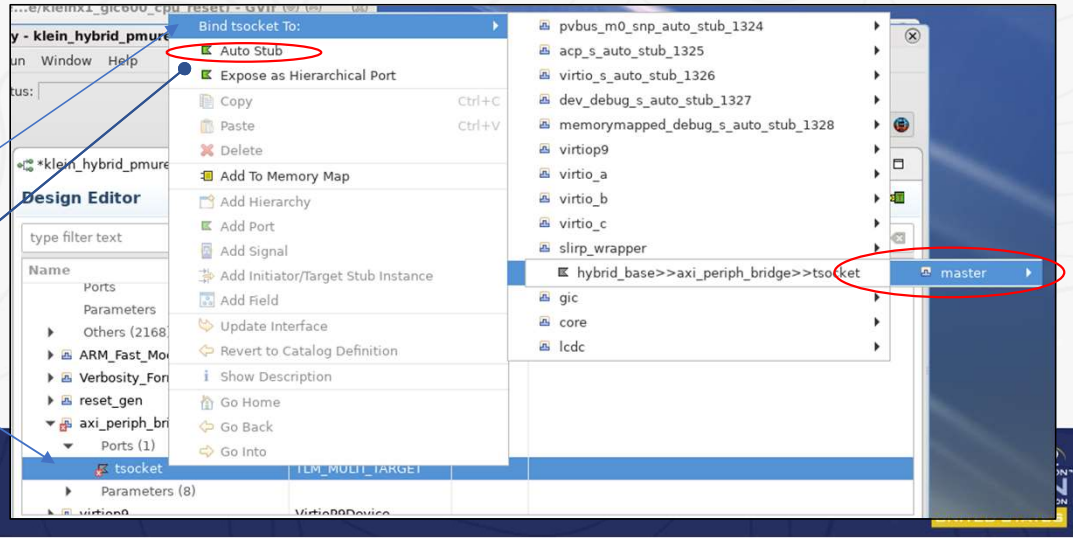
Method1: Bind Selected Port

- Port binding through ‘Ports’ tab.
- Select the unbound port in LHS, then the ports eligible to be bound will be shown in RHS. Select the port and right click and do ”Bind Selected Port”



Method2: Context Menu

- Right click on port → Bind Candidates → Select model → Select Port to bind



“Auto Stub” port if not required to connect anywhere.

Slide 20

AW9 Should mention the Auto stub capability here;
Andrew Wilmot, 11/15/2021

AB7 Yes. I will add call out and talk about it.
Ankit B, 11/16/2021

AB23 Added call out
Ankit B, 11/17/2021

Memory Maps in Helium Platform Assembly

- Create, View and Edit Memory Map using Memory Map View
- Memory Map creation is eased by automatic entry creation and automatic connections when using the Cadence router

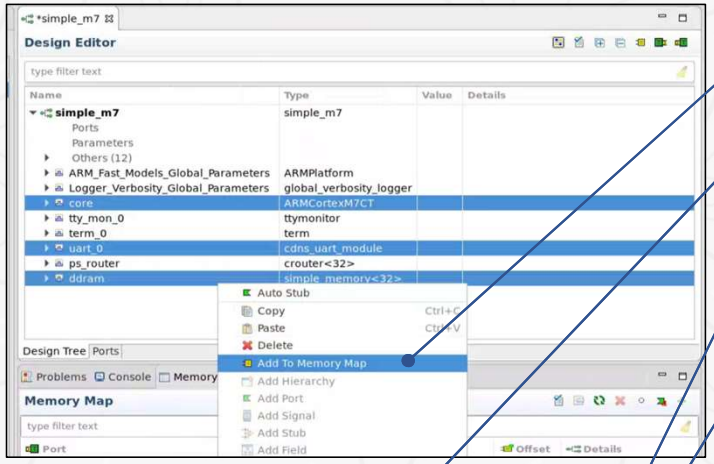
Memory Map

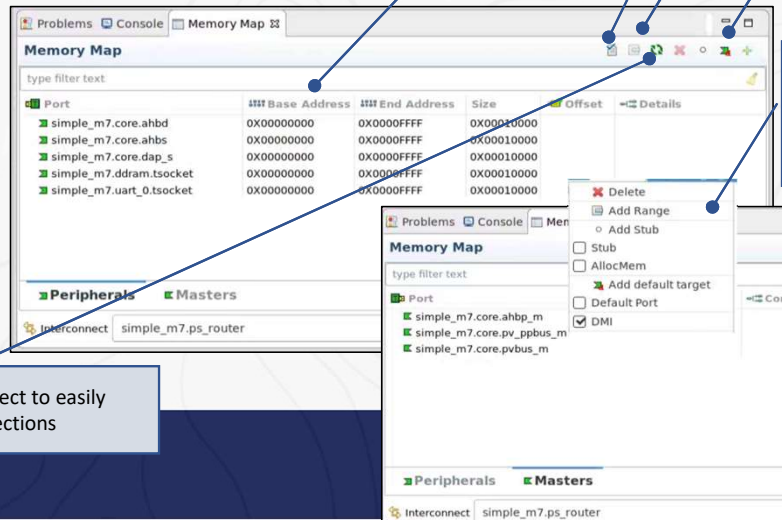
type filter text

Port	Base Address	End Address	Size	Offset	Details
stub_sram	0x04000000	0x0403FFFF	0x00040000		Stub, DMI, AllocMem
stub_unknown	0x1C0E8000	0x1C0E8FFF	0x00001000		Stub, DMI, AllocMem
stub_usb	0x1B000000	0x1B000FFF	0x00001000		Stub, DMI, AllocMem
vexpress_v9.aci.bus	0x1C040000	0x1C04FFFF	0x00010000		Connected
vexpress_v9.bridge.tsocket					Default, Connected
vexpress_v9.dualtimer_0.bus	0x1C110000	0x1C110FFF	0x00001000		Connected
vexpress_v9.dualtimer_1.bus	0x1C120000	0x1C120FFF	0x00001000		Connected
vexpress_v9.dualtimer_2.bus	0x1C130000	0x1C130FFF	0x00001000		Connected
vexpress_v9.dummy_sram.tsocket	0x2E000000	0x2E1FFFFF	0x00200000		Connected
vexpress_v9.ethernet.tsocket	0x1A000000	0x1A0000FF	0x00000100		^_auto_simple_buswidth_converter_26.isocket
vexpress_v9.gic.pvbus_s	0x2C000000	0x2CFFFFFF	0x01000000		^_auto_simple_buswidth_converter_22.isocket
vexpress_v9.lcdc.keyboard_pvbus_s	0x1C060000	0x1C060FFF	0x00001000		^_auto_simple_buswidth_converter_25.isocket
vexpress_v9.lcdc.lcdc_pvbus_s	0x1C1F0000	0x1C1F0FFF	0x00001000		^_auto_simple_buswidth_converter_23.isocket
vexpress_v9.lcdc.mouse_pvbus_s	0x1C070000	0x1C070FFF	0x00001000		^_auto_simple_buswidth_converter_24.isocket

Editing a memory map

- **a510_gic600_virtio_axi_hybrid** already instantiates crouter as an interconnect
- Easily add to memory map by multi-select of instances (initiators/targets) and ports
- "Validate" runs the Memory map rules checker
- **Reconfigure** generates the actual connections between router, targets and initiators





Add to Memory Map action

You can edit start, end, size and offset fields. Fields are recalculated based on edits.

Validate entry

Sub-ranges can be added for a selected entry using the "Add Range" action.

Default target can be added using the "Add Default Target" button or context menu.

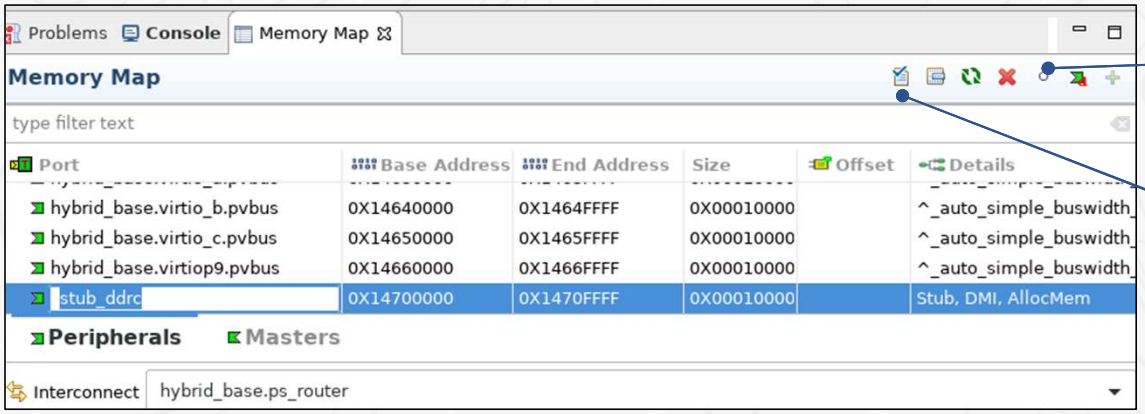
Other attributes of an entry can be modified using the context menu e.g. "DMI", "Stub" and "AllocMem"

Reconfigure and Connect to easily make necessary connections

Create & Remove RTL Stubs

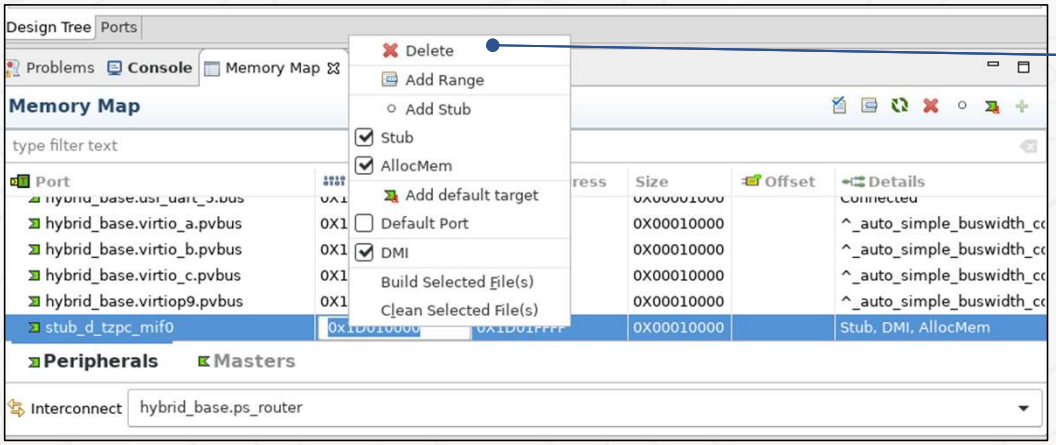
RTL Stub

- **RTL stub** can be added using ***"Add Stub"*** button in early phase of development, when specific **RTL IP** is not integrated in design
- Helium supports to add stubbing those RTL accesses and continue with bring up
- Remove the stubs when RTL IP is up in design and allow accesses to reach to IP



Click "Add Stub" button to create a stub in memory map, fill in name, start and end address.

Click on this button to validate that stub address is not overlapping with any other address



After integration with RTL design, the stub addresses used in the memory map will need to be removed. After Stub removal they will be by default routed to corresponding RTL IP

Helium Hybrid Adapters

- The Helium model catalog provides interface components (i.e. **Hybrid Adaptors**) to enable connections between:

Virtual Models ⇆ **RTL design**

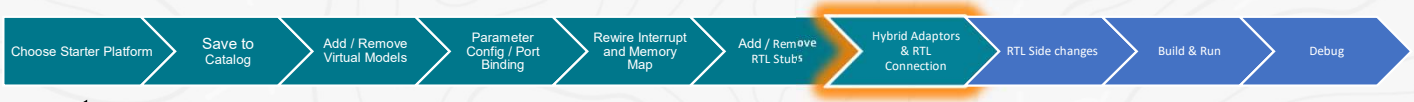
Helium

Xcelium

Palladium

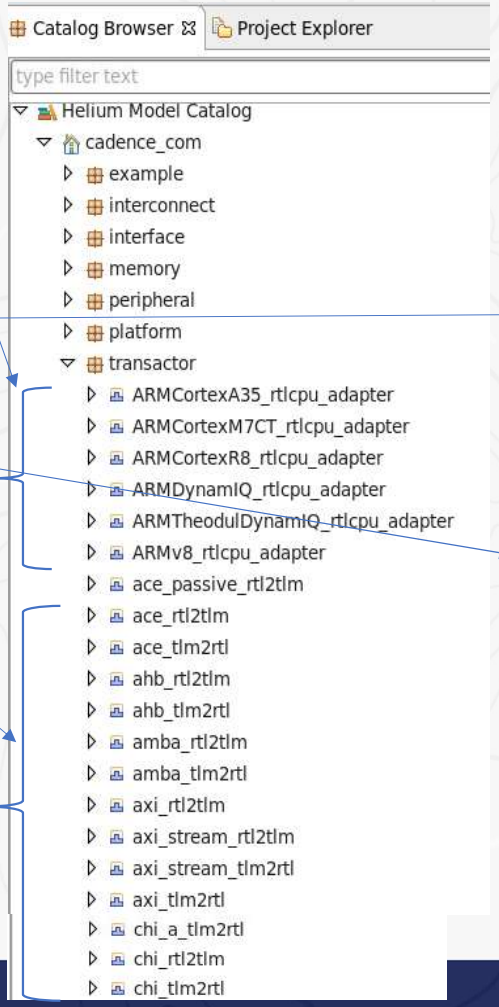
Protium

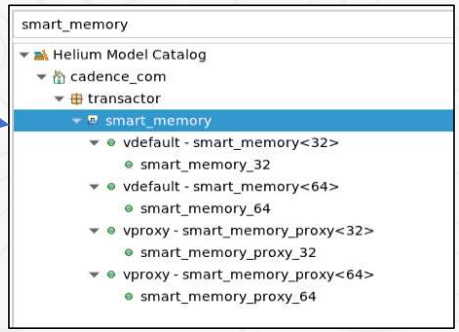
- Virtual side add/update Adaptor
- RTL side add/update Adaptor



Helium Hybrid Adapters

- **CPU Adapters**
  - Virtualize RTL cpu, pin-compatible with ARM FM core, connects seamlessly to RTL cpu wrapper using fast GFIFO communication channel
- **GIC Adapters**
  - Virtualize RTL GIC, pin-compatible with ARM FM GIC, connects seamlessly to RTL GIC wrapper using fast GFIFO communication channel
- **Smart memory**
  - Enables fast, concurrent, transparent view of memory to both virtual and RTL masters
- **Bus Protocol Adapters** (AXI, ACE, AHB, CHI)
  - Address RTL IP from virtual master, and vice versa, seamlessly connects with AVIPs
- **Base Signal Adapters**
  - Low level signal adapters for custom signal connection from TLM to RTL (sc_signal_tlm2rtl, sc_signal_rtl2tlm)





Search & Add virtual adapters for CPU and interrupt controller, Smart Memory
OR
Continue using from ref platform example and customize it based on need.

Configure RTL path in each Hybrid Adapter

- Configure **rtl_path** parameter –
  - Of each hybrid adapter to connect to appropriate hierarchy in RTL design

Design Editor

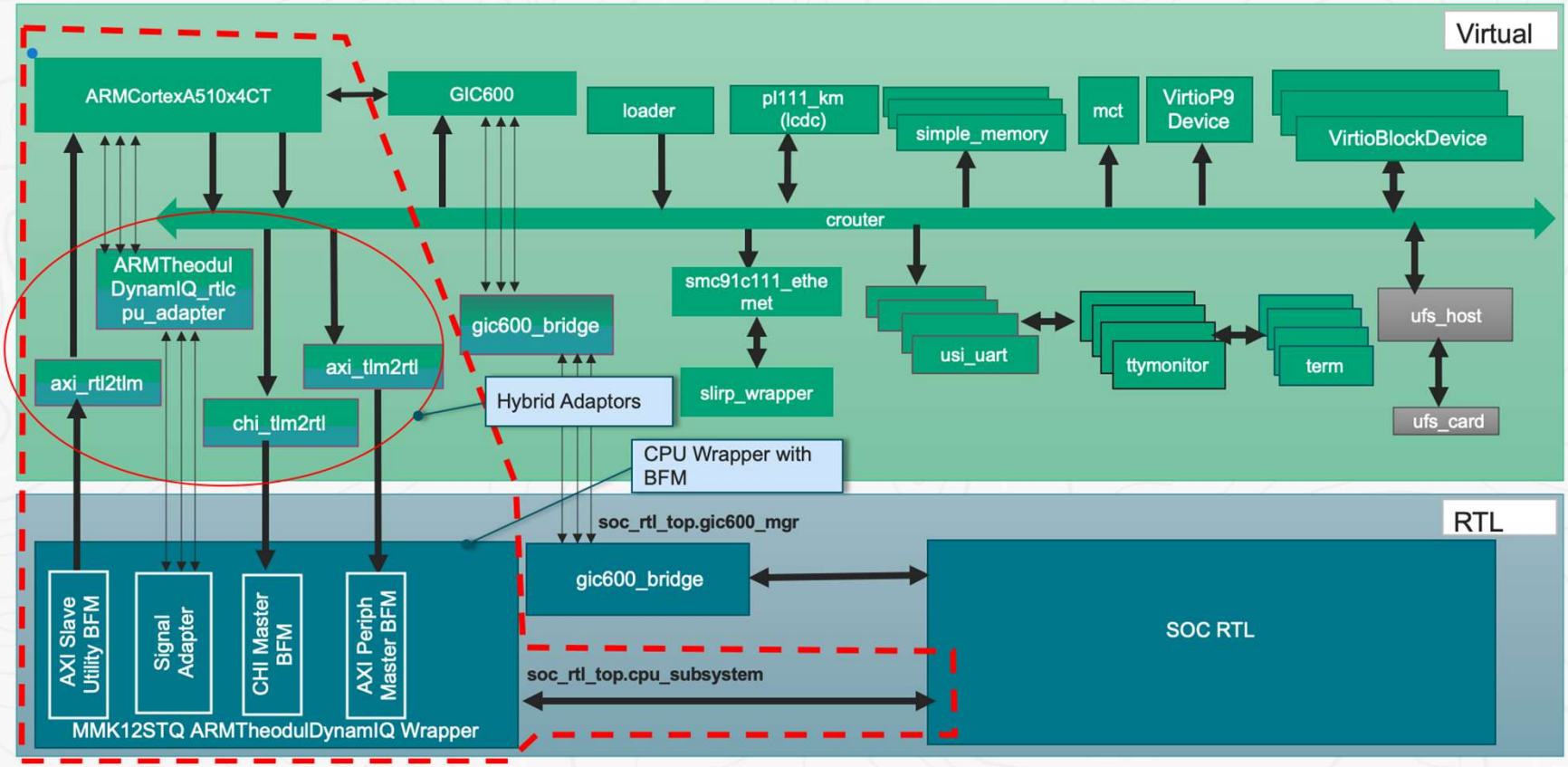
rtl_path

Name	Type	Value	Details
vexpress_v9	vexpress_v9		
gic600_bridge_inst	gic600_bridge		
Parameters (8)			
rtl_path	std::string	soc_rtl_top.gic600_mgr	Specify the hierarchical path of RTL ins
gic_mgr_axi	axi_rtl2tlm		
Parameters (6)			
rtl_path	std::string	soc_rtl_top.gic600_mgr.axis_bfm	Set the AVIP BFM RTL instance hierarch
rtlcpu_adapter	ARMTheodulDynamIQ_rtlcpu_adapter		
Parameters (4)			
rtl_path	std::string	soc_rtl_top.cpu_subsystem.cpu_sig_mgr	Set path of RTL cpu adapter in RTL hier
bridge	axi_tlm2rtl		
Parameters (8)			
rtl_path	std::string	soc_rtl_top.cpu_subsystem.axim_bfm	Set the AVIP BFM RTL instance hierarch
ddr_ram0	smart_memory<32>		
Parameters (32)			
rtl_path	std::string	soc_rtl_top.sm_mem_array	Set path of smart memory core in RTL

Filter with “rtl_path”

Provide the hierarchical path of RTL side of Adaptor

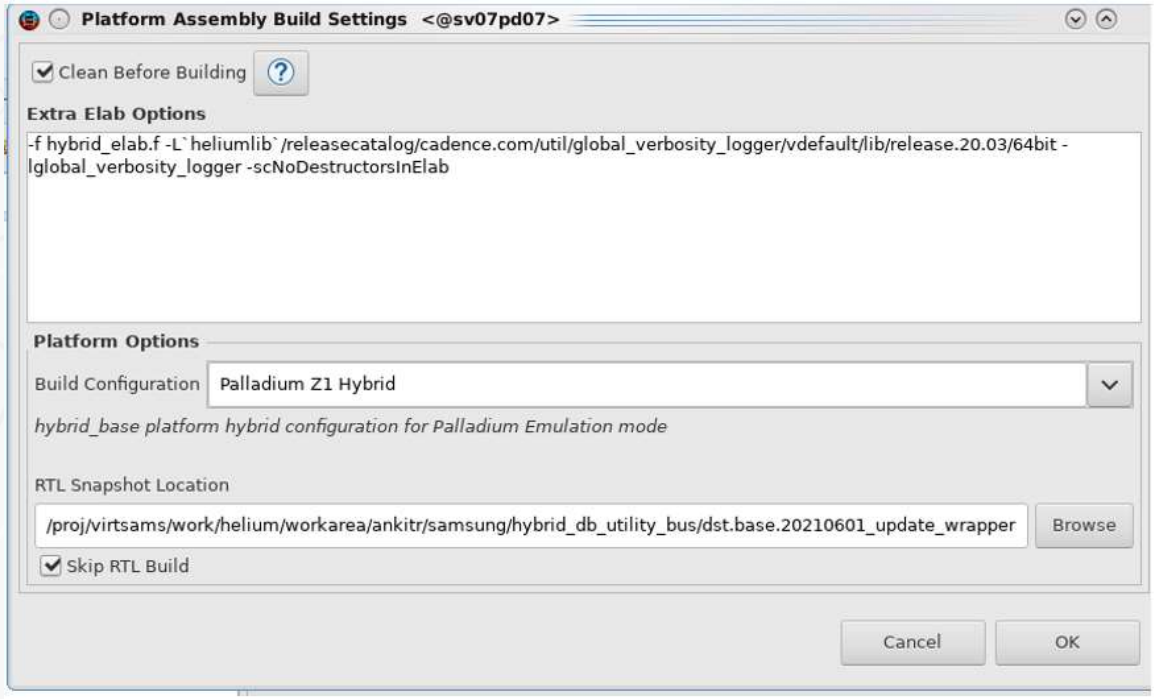
CPU Wrapper Connections in Example SoC Hybrid



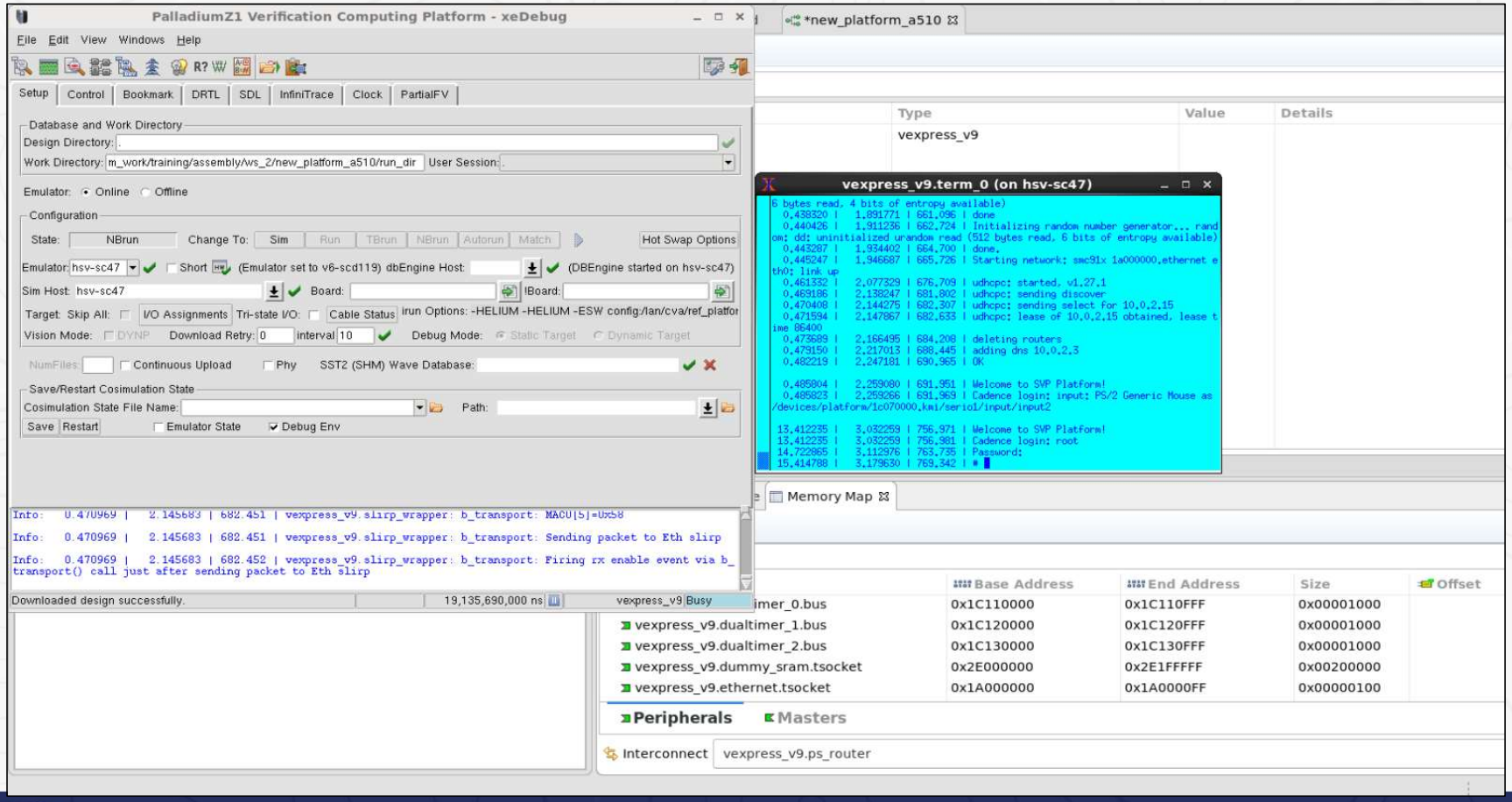
Platform Build Options – Hybrid Platform

- Click on **Project -> Configure Platform Build Options**
- Add the specific build options required for Platforms
- *Build Configuration*
  - 'Palladium Hybrid' – PZ1
  - 'Xcelium Hybrid' – Simulation
  - 'Protium X1 Hybrid' – Protium
- *Skip RTL Build*
  - If compiling RTL Snapshot separately

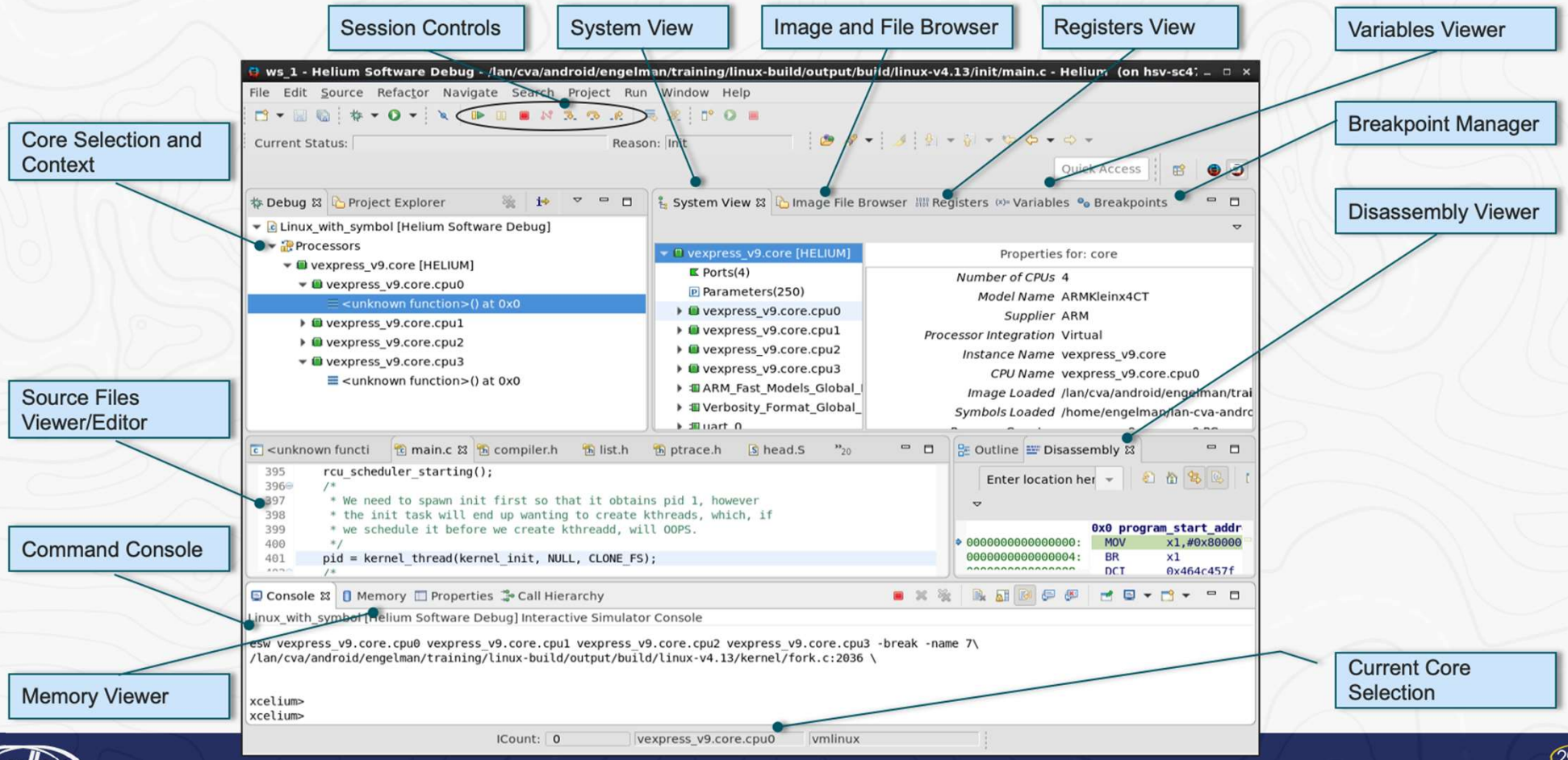
  Provide the path of *RTL Snapshot*



Run the Linux Demo - *a510_gic600_virtio_axi_hybrid* Platform



Debug Software with Integrated Debugger



Questions