COSIDE[®] SYNOPSYS[®] BRIDGE

ADVANCING
ARCHITECTUREJAN KASTNINGSYSTEMC AMS & COSIDE® UGM 2017
19 OCTOBER 2017





SECURE CONNECTIONS FOR A SMARTER WORLD

PUBLIC



Agenda

- 1. System design process
- 2. Architecture exploration process
- 3. Why advance?
- 4. COSIDE[®] Synopsys[®] Bridge
- 5. What's been done so far?
- 6. Simple proof of concept
- 7. What's next?
- 8. Accomplishments



System design process

- Top-down design structure
 - System decomposition
 - -Verify implementation and integration
 - -Validate system and product
- Achieve cost-efficient control over system design process
 - System optimization
 - Early identification of technical challenges

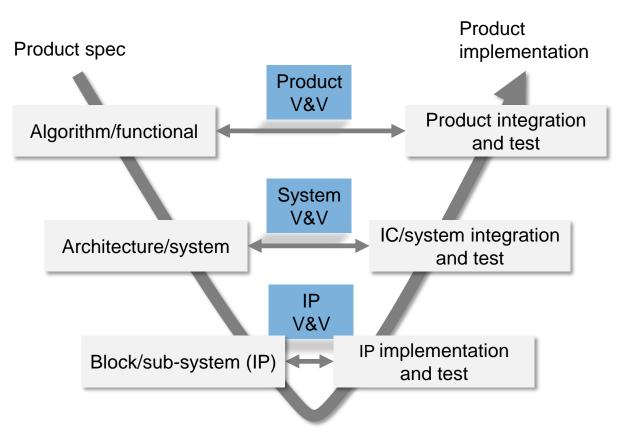
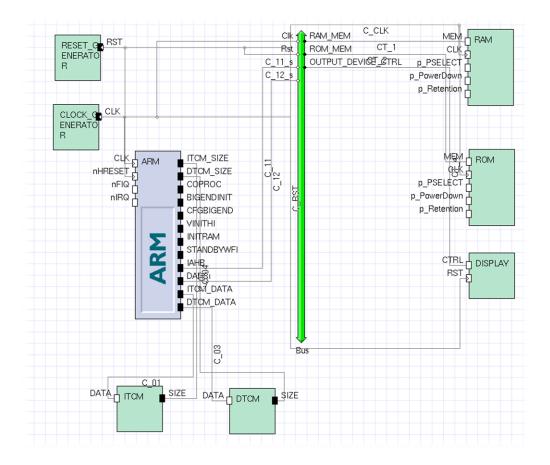


Illustration by Erwin de Kock, Martin Barnasconi



Architecture exploration process

Methodology



- Approach for system decomposition
- Estimate and optimize system resources
- Software: Platform Architect



Architecture exploration process

Model approaches

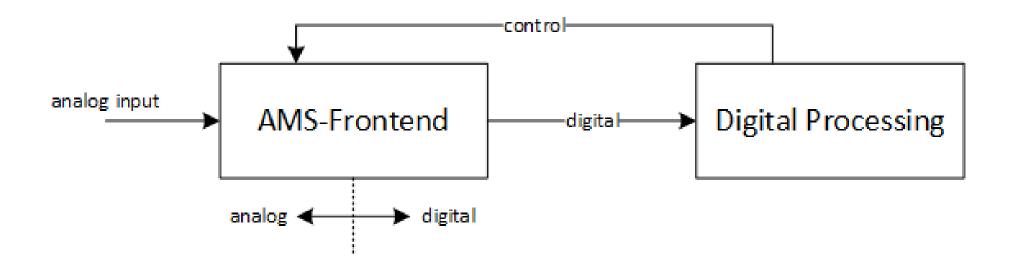
	High level	Low level
Design process	Architecture design / exploration	Block / sub-system design / exploration
Model base	Virtual processing / memory units → Virtual platform representation	IP models → Accurate platform representation
Simulation base	 Task graph based on: Resources (time & memory access) Task dependency → Basic behaviour 	 Input data Application images (source code) → Functional behaviour
Advantages	 Fast & early architecture exploration Sufficient for major design decisions 	 Cycle accurate decisions and task dependecies Allows fine tuning and design optimization
Outcome	Virtual platform which represents basic system behaviour	Highly accurate platform and functional behaviour representation
Use-case	Design decisions & system partitioning	System verification & optimization



Why advance?

Motivation

- Controlling AMS-IP behavior
 - Affects digital processing behavior
- Combine AMS-Frontend & digital processing





Why advance?

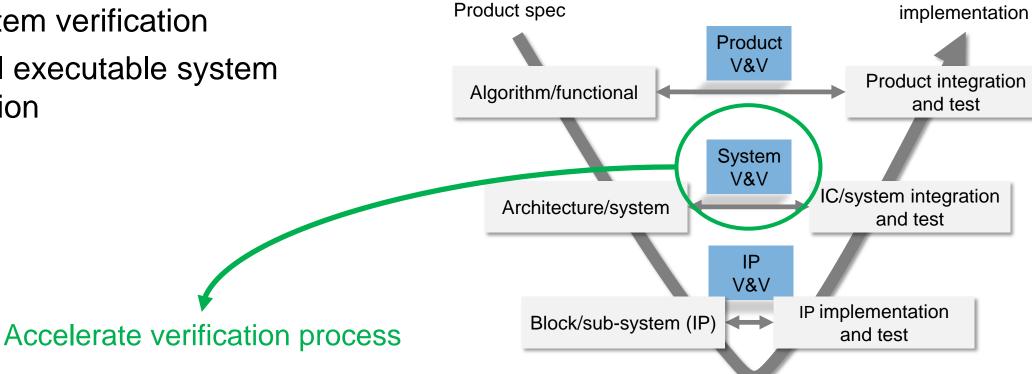
System complexity needs

- IP level
 - Input dependency and history
- System level
 - IP interconnections and relations
- AMS subsystems highly complex
 - Constraining or specify



Why advance? **Expectations**

- Better problem identification
- Early system verification
- High level executable system specification





Product

COSIDE® Synopsys[®] Bridge

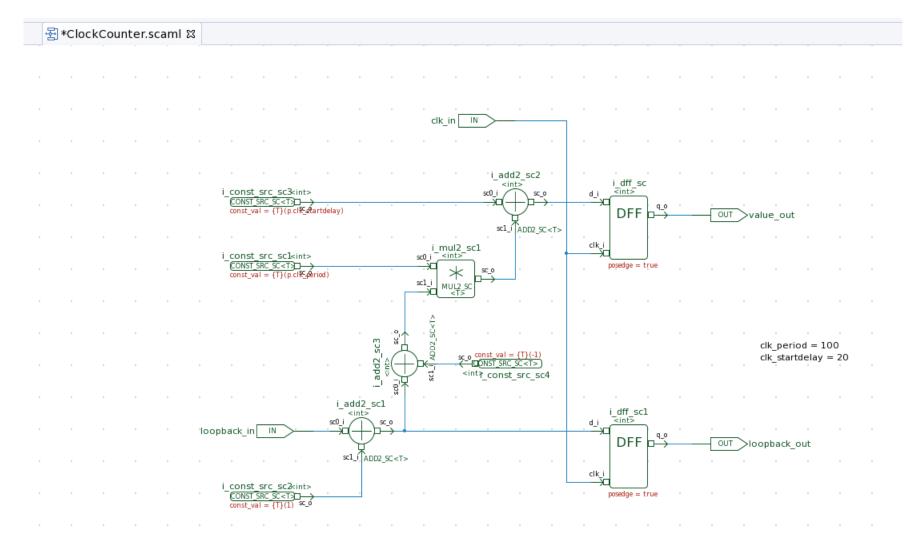
- Import any AMS model from COSIDE[®]
 - AMS model as standalone subsystem
 - Analog input controlled and defined in AMS model
 - Signal generator from COSIDE[®] library
 - Stimuli file import
- No tool configuration needed
 - Simple model import into Platform Architect
 - Basic SystemC port type support
- Full COSIDE[®] library support



What's been done so far?

- Analysing need of dynamic tool connection
- Determining use-cases
- Brainstorming possible solutions
 Getting Coseda & Synopsys onboard
- Starting tool development
- Testing and improving
- Simple proof of concept







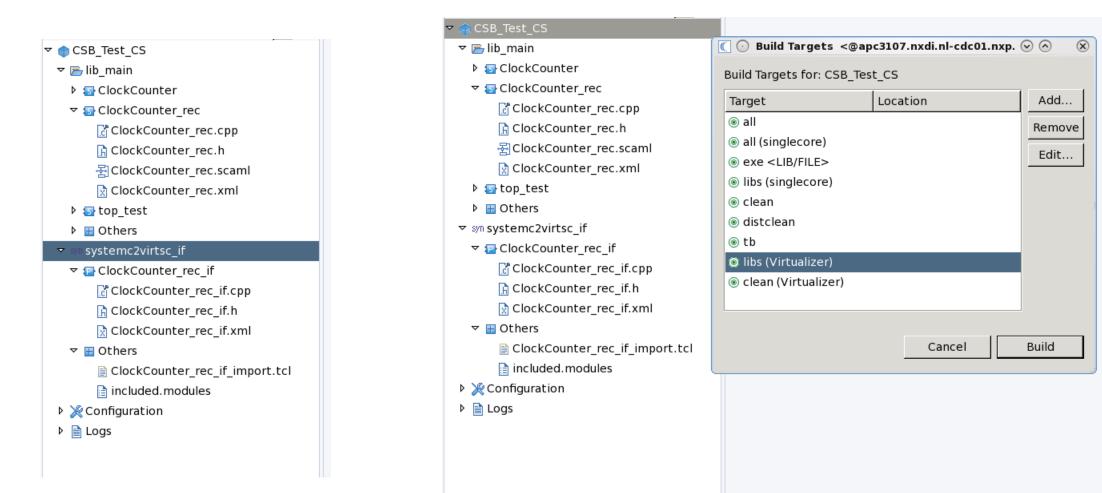
10 PUBLIC

쥠 *ClockCounter_rec.scaml 업	
	i_file_out_sc2
	FILE_OUT_SC
clk in	i_ClockCounter1
	separator = '
	Ioopback_in CLOCKCOUNTER format_int = file_out_sc<{T}>::DEC format_float = file_out_sc<{T}>::GENERAL precision = 15
	clk_period = (int) p.clk_period i i_file_out_sc1 file_format = file_out_sc<{T}>::ASCII clk_startdelay = (int) p.clk_startdelay <int></int>
	FILE_OUT_SC
	fname = "loopback_out.dat" fheader = "%time signal"
	store_time = true
	cik_period = 100 cik_startdelay = 20 format_int = file_out_sc<{T}>::DEC format_float = file_out_sc<{T}>::GENERAL
	precision = 15file_format = file_out_sc<{T}>::ASCII



r 🏟 CSB_Test_CS					
포 📂 lib_main					
ClockCounter					
マ 🔄 ClockCounter_rec					
ClockCounter_rec.cpp					
🔓 ClockCounter_rec.h					
唇 ClockCounter_rec.scaml					
🔀 ClockCounter_rec.xml			1		
▶ 5 top_test	New	• •			
🕨 🔠 Others	<u>O</u> pen	F3			
Configuration	Open Wit <u>h</u>	<u> </u>			
Logs	🖬 SystemC-AMS Module	•		-	
- 1	SystemC-AMS Tools	<u> </u>	🔓 SystemC-Header (*.h)		
	🗎 Сору	Ctrl+C	SystemC-Template (*.cpp)		
	🔓 Paste	Ctrl+V	🕜 SystemC-Header (*.h), Template (*.cpp) 🗈 Create Documentation		
	++ Move		쥠 Create Initial Schematic		
	I Rename	F2			
	× Delete	Delete	Generic Testbench Generic Testbench ■		
	🖨 Refresh	F5	·		
	ڬ Import		SystemC-AMS Couplings		
	🛁 Export		✓ SystemC-AMS Import ✓ SystemC-AMS Export	cā Incisive/Xcelium	
	<u>R</u> un As	•	·	syn Virtualizer	Create Interface
	<u>D</u> ebug As	►	Reset Symbol Graphics		
	<u>P</u> rofile As	►			
	Topcased	►			
	💖 Run <u>C</u> /C++ Code Analysis				
	T <u>e</u> am	►			
	Comp <u>a</u> re With	►			
	Rep <u>l</u> ace With	•			
	Properties	Alt+Enter			
E					







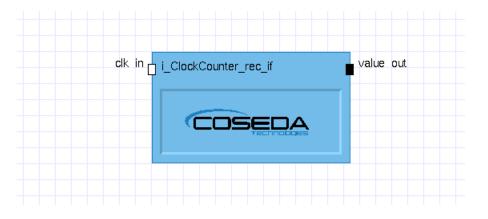
Tools View Simulation Plugins Help	🔒 📀 Execute Script <@apc3107.nxdi.nl-cdc01.nxp.com> 🛛 😔 🔗 🛛 🛞
Import Architecture File	Look in: 🔄 cuments/coside2.4RC1_WS/CSB_Test_CS/ 🗸 🗢 🗈 💣 🏢
Export Architecture File	
Export Memory Map Header File	
MCO +	
Propagate Port Properties	systemc2virtsc_if
Execute Script	
Manage Main Configurations	B ○ Execute Script <@apc3107.nxdi.nl-cdc01.nxp.com> = ○ ○ ※
Convert to Bridge	Look in: 🔄 IRC1_WS/CSB_Test_CS/systemc2virtsc_if/ 🔻 🔄 😷 🏢 🏢
Convert to HW Interface Block	
Convert to SW Driver	File type: Script Files (*.tcl)
Convert to Channel	
Convert to Transactor	
Revert to HW Interface Block	i obj-virt-4.8 ClockCounter rec if import.tcl
Revert to Block	
	File name: ClockCounter_rec_if_import.tcl
	File type: Script Files (*.tcl) Cancel



Console							
= 11	= Copyright(R) COSEDA Technologies GmbH. All rights reserved. = COSEDA - Synopsys Bridge (CSB) - Version: 1.1 = support: support@coseda-tech.com = phone: +49 351 32149080						
	Starting COSIDE-Virtualizer Bridge (CSB) for module "ClockCounter_rec_if" executing "ClockCounter_rec_if_import.tcl" Starting COSIDE-Virtualizer Bridge (CSB) for module "ClockCounter_rec_if" executing "ClockCounter_rec_if_import.tcl" SB::INFO: import done						
Console	Messages	Build Output	Parameter Editor	Memory Map Table	Memory Maps	Constraint Editor	

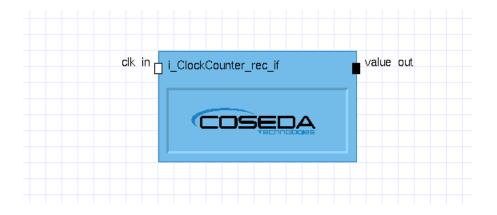
Definitions	QuickRef	
🚊 わ Project		
🖻 🖪 Cloc	kCounter_re	ec_if
÷- 🕨 🤇	CSB_Test_C	S_namespace::ClockCounter_n
-+⇒CLC		
— 🖛 Defa	ault	
— 🖛 Defa	ault UT	
t ⇒ Defa	ault_indexed	UT
🖛 REN	1AP	-
l l	ΈT	
±- External D	efinitions)	





Name 🗸	Value	Configuration	Visibility	Editability	
🚊 🗟 Block properties					
- Name	i_ClockCounter_rec_if				
🖻 🔝 Scml Properties					
- clk_period	100	Default	 Visible 	 Until Simulation Start 	
🦾 = clk_startdelay	20	Default	 Visible 	- Until Simulation Start	
Console Messages I	Build Output Paramete	r Editor Memory Ma	ap Table 🛛 Mem	ory Maps Constraint E	Editor

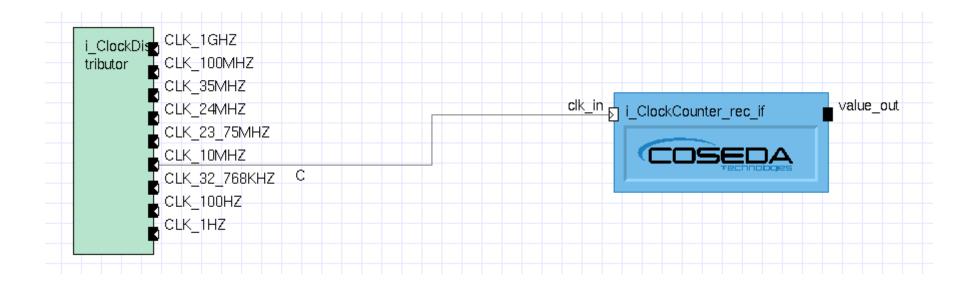




pct.protocol = CLOCK

arameters - /HARDWARE/i_ClockCounter_rec_if/clk_in							
Name 🗸	Value	Visibility	Editability				
⊡- ⊿ ⊢clk_in							
🗄 📑 Port Properties							
Direction	In						
— — MasterSlaveness	Slave						
Category	Clock						
🛶 Protocol	CLOCK						
Console Messages Build Output P	arameter Editor	Memory Map	Table Memory Maps Constraint Editor				







🗢 🔷 Current Simulation (suspended at 0:00:00	Item	Offset	Value
🗢 🐫 Design Hierarchy	▶ clk_in		0×00000000
▼ ∎ HARDWARE	Ioopback_in		0
マ ∎ i_ClockCounter_rec_if	Ioopback_out		0
マ ∎ i_ClockCounter_rec	value_out		0
▼ 🛚 i_ClockCounter1	⇔ unnamedNet1		0/0
i_add2_sc1	⇔ unnamedNet2		100/100
i_add2_sc2	⇔ unnamedNet3		1/1
i_add2_sc3	⇔ unnamedNet4		0/0
i_const_src_sc1	⇔ unnamedNet5		0/0
i_const_src_sc2	\leftrightarrow unnamedNet6		0/0
i_const_src_sc3	⇔ unnamedNet7		0/0
i_const_src_sc4	\leftrightarrow unnamedNet8		-1/-1
i_dff_sc	▶ o p		{}
<pre>i_dff_sc1</pre>	◊ C		0x2976878
<pre>i_mul2_sc1</pre>			:
i_file_out_sc1			
i_file_out_sc2			
i_ClockDistributor			



🗸 🔷 Current Simulation (suspended at 0:00:00	Item	Offset	Value
🗢 斗 Design Hierarchy	▶ clk_in		0×00000001
▼ ∎ HARDWARE	Ioopback_in		2
マ ∎ i_ClockCounter_rec_if	Ioopback_out		2
▼ 🏽 i_ClockCounter_rec	value_out		100
▼ 🛛 i_ClockCounter1	⇔ unnamedNet1		3/3
i_add2_sc1	⇔ unnamedNet2		100/100
i_add2_sc2	⇔ unnamedNet3		1/1
i_add2_sc3	⇔ unnamedNet4		200/200
i_const_src_sc1	⇔ unnamedNet5		0/0
i_const_src_sc2	⇔ unnamedNet6		200/200
i_const_src_sc3	⇔ unnamedNet7		2/2
i_const_src_sc4	⇔ unnamedNet8		-1/-1
i_dff_sc	▶ o p		{}
<pre>i_dff_sc1</pre>	◊ C		0x2976878
<pre>i_mul2_sc1</pre>			
<pre>i_file_out_sc1</pre>			
i_file_out_sc2			
i_ClockDistributor			



What's next?

- Full-blown proof-of-concept
 - AMS functionality
 - Coside library building blocks
- TLM register support



Accomplishments







Thomas Arndt

Martin Barnasconi

Christian Schuster

Karsten Einwich

Martin Klein

Stefan Thiel





SECURE CONNECTIONS FOR A SMARTER WORLD

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2016 NXP B.V.