

# Developing The Next Generation LiDAR with Model-Based Design

Chenji TU, Valeo Detection Systems GmbH



MathWorks AUTOMOTIVE CONFERENCE 2024

#### Agenda

- Overview of Valeo's ADAS Portfolio and LiDAR (Scala<sup>™</sup>) Evolution
- Motivation to adopt Model-Based Design (MBD)
- LiDAR Development with MBD
- Sensor's Azimuth Accuracy, Precision, Missing Shot Improvement
- FPGA Area, Speed Optimization and Verification with MBD
- Key Takeaways
- Gratitude to MathWorks Professionals and Consultants

### Overview of Valeo's ADAS Portfolio and LiDAR (Scala<sup>™</sup>) Evolution

Valeo, a mobility leader, adopts Model-Based Design (MBD) for advanced LiDAR in automotive ADAS.

LIDAR



MathWorks AUTOMOTIVE CONFERENCE 2024

#### Motivation to adopt Model-Based Design (MBD)

**MBD** reduces time to market and design cost in a multidisciplinary, exponentially complex safety critical development process.



#### MBD is the key to efficiency

#### MathWorks AUTOMOTIVE CONFERENCE 2024

#### LiDAR Development with MBD – Lidar System Overview

#### Overview of LiDAR System Elements and Signal Chain.



LiDAR point cloud begins with motor driven laser shots

#### LiDAR Development with MBD – SPAD

Temperature Dependent Detector Sensitivity Feedback Loop Control Development is based on **Simscape** (Thermal Port), **DSP** (Noise Floor), **Control System Toolbox** (PI).

Quick system behavioral modeling, simulation, and auto code generation for controller

- Diode, Power Supply, DAC,...
- Temperature-dependent plant and setpoints
- Controllability scenarios
- Delay, noise floor, SNR,...
- PI controller, LPF targeting ARM





### LiDAR Development with MBD – Laser Power Management

Laser Power Estimation Development (Range, Eye Safety) is based on **Simulink** (Signal Profile, Detection, High-Speed Sampling) and **MATLAB, Statistics Toolbox** (Correlation, Algorithm).

Quick System Behavioral Modeling, Simulation and Algorithm Evaluation:

- Optical Gaussian pulse
- Xilinx differential input buffer
- DAC threshold swiping
- Time-to-digital conversion
- Pulse profile reconstruction
- Correlation, power estimation







### LiDAR Development with MBD – Motor Control

Motor Control improvement investigation is based on **SimScape** (BLDC, SVPWM, Gate Driver) and **Control System Toolbox** (Control Algorithms).

Quick system behavioral modeling, simulation, and Algorithm Evaluation:

- Investigate potential improvements in existing motor system design for speed regulation stability, position control, noisy APS, etc.
- Collaborating with MathWorks consulting services





### LiDAR Development with MBD – Motor Angle Decoder & Laser Driver

Motor Angle Decoder and Laser Driver development is based on HDL Coder, Stateflow, Simulink,...



#### MathWorks AUTOMOTIVE CONFERENCE 2024 Sensor's Azimuth Accuracy, Precision, Missing Shot Improvement I

**MBD** is the cornerstone in development, from ideas to algorithms with automatic production code generation, ISO / ASPICE process compliance till KPI / requirements fulfillment of the product.



#### MathWorks AUTOMOTIVE CONFERENCE 2024 Sensor's Azimuth Accuracy, Precision, Missing Shot Improvement II

Easy to use tools and Apps in an integrated, consistent development environment with MBD



### FPGA Area and Speed Optimization with MBD

- Fixed-Point Designer detects overflows and potential errors, proposes HW-efficient data types
- Lookup Table Optimizer maps arithmetics to RAM, helps with trade-off exploration and design optimization



Passed Implementation.				📣 Lookup Table Optimizer		Passed Implement	ation.		1		
Parsed resource report file: <u>RPM_2_RT_Calc</u>			RT Calc o	Objective	Objective >		Parsed resource report file: <u>RPM_2_RT_Calc_utilization_placed.rp</u>				
Resource summa	arv	1				Resource summa	ary				
Resource	Usage	A	TAT	Area imp	roves 5.3x 📥 🖊	Resource	Usage	2 wailable	Utilization (%)		
CLB LUTs	887	44000	2.02	1-D T(u)		CLB LUTs	167	4000	0.38		
CLB Registers	116	88000	0.13	> / > Welcome to the L	ookup Table (LUT) Optimizer	CLB Registers	128	88000	0.15		
DSP	0	400	0.00						1.25		
Diants DAM Tite	0	100	0.00	Select the source for memory	Select the source for memory efficient LUT				0.50		
BIOCK KAM THE	0	100	0.00								
Parsed timing repo	ort file:	timing_pos	t_route.rpt	Simulink block or subsyster     Create a memory efficient LUT fi	n or a Simulink block	Parsed timing repo	ort file: <u>t</u> v	iming_post	route.rpt.		
Timing summar	y						Value				
	Value	•		MATLAB Function Handle		Requirement 4 ns (250 MHz)					
Requirement	4 ns (	250 MHz)		Create a memory efficient LUT fi	or a MATLAB function handle	Data Path Dalay	2,222				
Data Path Delay	22.83	1 ns				Slack	0.719	ns			
Slack	-18.85	51 ns		O Fitted (	oves 304.8x	Clock Frequency	304.79	MHz			
Clock Frequency	43.76	MHz		Create a memory efficient LUT fr	or a curve fit object from base workso	308					
WARNING: Timir	ngs not	met.									



#### MathWorks AUTOMOTIVE CONFERENCE 2024

#### FPGA Area and Speed Optimization with MBD - Example

**Example:** The original design of Scan Angle for Point Cloud Data Structure has overflow and consumes many FPGA Resources.

- Overflow detected in design
- Consumes many CLB LUTs

🝌 SA32_Gen_NEW_viva	o - [E:/TU/pjWORKSPACE/pjAASC/code/SA32_Ref/240126_1138/vivado_prj/SA32_Gen_NEW_vivado.xpr] - Vivado 2022.1								
<u>File Edit Flow</u>	Tools Reports Window Layout View Help Q- Quick Access								
	■ m × ∞ ▶, # 51 ℃ ⊉ Σ ≤ Ø Ø								
IMPLEMENTED DES	GN - xcau10p-sbvb484-1-i								
Tcl Console Me	ssages Log Reports Design Runs Utilization × Timing Power Methodology DRC Pac	age Pins I/O I	Ports						
origato	Q ≚ ♦ % Hierarchy		-						
Hierarchy Summary	Name	CLB LUTs (44000)	CLB Registers (88000)	CARRY8 (9720)	F7 Muxes (38880)	CLB (9720)	LUT as Logic (44000)	LUT as Memory (40320)	DSPs (400)
✓ CLB Logic	✓ N SA32_Gen_NEW	1633	1623	49	1	372	1470	163	12
CARRY8 (1	u_nfp_add_comp (nfp_add_single)	505	532	17	1	129	464	41	0
F7 Muxes (<	u_nfp_relop_comp (nfp_relop_single)	4	12	0	0	4	4	0	0
✓ CLB LUTs (	u_ShiftRegisterRAM (SimpleDualPortRAM_generic)	20	17	0	0	8	0	20	0
✓ LUT as	u_ShiftRegisterRAM_generic (SimpleDualPortRAM_genericparameterized1)	10	16	0	0	7	0	10	0
LUT	u_tbSA32_SA32_Gen_NEW_nfp_convert_sfix_75_En37_to_single (nfp_convert_sfix_75_En37_to_single)	) 521	468	17	0	126	519	2	0
LUT	u_tbSA32_SA32_Gen_NEW_nfp_convert_single_to_fix_32_En0 (nfp_convert_single_to_fix_32_En0)	345	214	7	0	85	321	24	0

### FPGA Area and Speed Optimization with MBD – Fixed-Point Tool I

Use the App "**Fixed-Point Tool**" to convert the data types used in the design under optimization to more efficient ones, i.e., given a defined tolerance, to analyze over/underflow in the signal chains, to automatically propose improved data types, leading to reduced Area.

(1) Open Fixed Point Tool App





Select a Workflow           Optimized Fixed-Point Conversion	
Iterative Fixed-Point Conversion         Iterative Fixed-Point Conversion           Automatically propose fixed-point data types and manually select which data types to apply to your model.           Example         Help	
Range Collection ③ Explore numerical behavior of your model pre- or post-conversion. Example Help	

MathWorks AUTOMOTIVE CONFERENCE 2024

#### (3) Prepare to create restore point



(4) To automatically collect signals' data ranges driven by a full range test vector

# FPGA Area and Speed Optimization with MBD – Fixed-Point Tool II

Step-by-Step follow the intuitive flow in the App

#### (5) User's inputs directive for Optimization Tool's searching process

🕞 Settings 👻	No.		2				
PROPOSE				W			
Propose:			Fraction Length 🔹				
Propose signedness:			Yes	*			
Safety margin for simulation min/max (%):							
CONVERT TO FIXED POIN	т						
Convert double/single/ha	lf typ	es:	Yes 🔹				
Convert inherited types:			Yes 🔹				
Default word length:			32				
Default fraction length:			4				
Original Data Type		Word Leng	<u>yth</u>	Fraction Length			
Double/Single/Half		32		Will propose			
Inherited		32		Will propose			
Fixed point		No chano	je i	Will propose			

(6) Tool automatically detects critical Overflow and Underflow. Evaluate. Go back to (5) and change directive, or analyze then change simulink design to solve critical issues.

Bias1						
	single		n/a		-12883703808	8589934592
Ch_I_Rising_Edge	boolean	boolean	n/a			
Cnt_M		fixdt(1,16,0)	fixdt(1,16,0)			
Config_MirrorSideOffset		fixdt(1,16,0)	fixdt(1,16,0)			
Constant1		fixdt(1,16,0)	fixdt(0,16,3)			
Constant18		fixdt(1,16,0)	fixdt(0,16,4)			
Constant2		fixdt(0,55,34)	fixdt(0,55,35)			
Constant3		fixdt(0,55,34)	fixdt(0,55,34)			
Date Type Conversion2	bol((0.32(0)	ikdt(0.32.0)	fixd((0,32,0)		0	322122547
Data Type Conversion3	single	single	fixdt(1,32,-3)	✓	-12883703808	429496729
Delay1	boolean		n/a			
Delay25	boolean		n/a			
Delay35			n/a			
Delay36	boolean		n/a			
Delay37	boolean		n/a			
Delay40			n/a			
ExtrapolatedTick_En_i	boolean	boolean	n/a			
Gain8	fixdt(1,67,37)	fixdt(1,67,37)	fixdt(1,67,37)		-392675328	392663344
Gain8 : Gain		Inherit: Inherit via inter	. n/a			
ualization of Simulation Data						
		Histograms of all resu	Its in the model			
<b>A</b>						
222						
232						
	CriM Config_MirrorSideOffset Constant1 Constant1 Constant2 Constant3 Delay1 Delay25 Delay36 Delay37 Delay40 ExtrapolatedTick_En_i Gain8 Gain8: Gain	Crit_M Config_MirroSideOffset Constant1 Constant1 Constant1 Constant2 Constant3 Delay1 Delay25 Delay36 Delay36 Delay36 Delay37 Delay36 Delay36 Delay37 Delay36 Delay36 Delay37 Delay36 Delay36 Delay36 Delay37 Delay36	Cm_M     itedati(1:6.0)       Config_MirroSideOffset     fixdt(1:6.0)       Constant1     fixdt(1:6.0)       Constant18     fixdt(1:6.0)       Constant2     fixdt(0:55.34)       Constant3     fixdt(0:55.34)       Data Type Conversion3     single       Delay1     boolean       Delay25     boolean       Delay36     boolean       Delay37     boolean       Delay37     boolean       Delay36     boolean       Gain8     fixdt(1:6.7,37)       Gain8     fixdt(1:6.7,37)	Cht, M         Itxd(1,16,0)         Itxd(1,16,0)         Itxd(1,16,0)           Config_MirroSideOffset         fixd(1,16,0)         fixd(1,16,0)         fixd(1,16,0)           Constant1         fixd(1,16,0)         fixd(1,16,0)         fixd(1,16,0)           Constant18         fixd(1,16,0)         fixd(1,16,0)         fixd(1,16,0)           Constant18         fixd(1,16,0)         fixd(1,16,0)         fixd(1,16,0)           Constant2         fixd(1,16,0)         fixd(1,05,34)         fixd(1,05,34)           Delarstrand         fixd(1,0,0)         fixd(1,0,0,34)         fixd(1,0,0,34)           Data Type Conversion3         single         single         fixd(1,16,2,-3)           Delay 1         boolean         n/a         n/a           Delay25         boolean         n/a         n/a           Delay36         boolean         n/a         n/a           Delay37         boolean         n/a         max           Delay36         boolean         n/a         max           Delay36         boolean         n/a         max           Delay36         boolean         n/a         max           Delay36         boolean         n/a         max           Delay36	Cnt_M     ixxdt(116.0)     ixxdt(116.0)       Config_MirrorSideOffset     fixdt(116.0)     fixdt(116.0)       Constant1     fixdt(116.0)     fixdt(116.0)       Constant18     ixxdt(116.0)     fixdt(0.16.3)       Constant18     ixxdt(116.0)     fixdt(0.16.3)       Constant2     fixxdt(0.55.34)     fixxdt(0.55.34)       Constant3     ifxxdt(0.55.34)     fixxdt(0.55.34)       Data Type Conversion3     single     single       Data Type Conversion3     single     single       Delay25     boolean     n/a       Delay36     boolean     n/a       Delay37     boolean     n/a       Delay37     boolean     n/a       Gain8     fixdt(1.67.37)     fixdt(1.67.37)       Gain8     fixdt(1.67.37)     fixdt(1.67.37)	Cht_M         ixxd(1,16,0)         Ixxd(1,16,0)

(7) Let App propose new fixed point data types in the design, then change the design with new types, fix some minor type mismatch errors, then Simulate, compare the differences, analyze,...



	5
Compare Results	Restore Original Mor
i ile series	MANAGE



### FPGA Area and Speed Optimization with MBD – Fixed-Point Tool III

(8) Iteratively develop testbench to cover full range inputs and to evaluate performance toleration



MathWorks AUTOMOTIVE CONFERENCE 2024

# FPGA Area and Speed Optimization with MBD – Fixed-Point Tool IV

#### Compare Utilization before and after "Fixed-Point Tool" workflow:

HDL Workflow Advisor - tbSA32/SA32_Gen_NEW		HDL Workflow Advisor - tb_SA_Single_Gen/SA_Single	_Gen
File Edit Run Help		File Edit Run Help	
Find: V 🗢		Find: V V	
<ul> <li>Find: </li> <li>HDL Workflow Advisor</li> <li>Set Target</li> <li>A 1.1. Set Target Device and Synthesis Tool</li> <li>1.2. Set Target Frequency</li> <li>2. Prepare Model For HDL Code Generation</li> <li>2.1. Check Model Settings</li> <li>3. HDL Code Generation</li> <li>3.1. Set HDL Options</li> <li>A 3.2. Generate RTL Code and Testbench</li> <li>Set Appendent Project</li> <li>4.1. Create Project</li> <li>4.2. Perform Synthesis and P/R</li> <li>4.2.1. Run Synthesis</li> <li>4.3. Annotate Model with Synthesis Result</li> </ul>	4.2.2. Run Implementation         Analysis         Run place and route for specified FPGA device         Input Parameters         Skip this task         Ignore place and route errors         Run This Task         Result:  Passed         Passed Implementation.         Parsed resource report file: SA32_Gen_NEW_utilization_placed.rpt.         Resource       Usage         Available       Utilization (%)         CLB LUTs       1633	Find:       Image: Constraint of the second se	<b>4.2.2. Run Implementation</b> Analysis         Run place and route for specified FPGA device         Input Parameters         Skip this task         Ignore place and route errors         Run This Task         Result:       Implementation.         Passed Implementation.         Parsed resource report file:       SA_Single_Gen_utilization_placed.rp         Resource summary       Resource         Resource       Usa te Available         Utilization (%)       CLB LUTs         CLB Registers       636
	CLB Registers       1623       88000       1.84         DSPs       12       400       3.00         Dted. RAMETIL:       100       0.00         URAM       0       0         Parsed timing report file:       timing_post_route.rpt.         Timing summary       Requirement       4 ns (250 MHz)         Data Path Delay       3.54 ns         Slack       0.403 ns         Clock Frequency       278.01 MHz         Generated Post Route Timing Report timing_post_route.rpt.         Task "Run Implementation" successful.         Generated logfile:       E:\TU\pjWORKSPACE\pjAASC\code\SA32_Ref\240124	s 54% reduced	DSPs       9       400       2.25         Biock RAW The       0       100       0.00         URAM       0       0       0         Parsed timing report file: timing_post_route.rpt.       Timing summary         Requirement       4 ns (250 MHz)         Data Path Delay       3.863 ns         Slack       0.118 ns         Clock Frequency       257.60 MHz         Generated Post Route Timing Report timing_post_route.rpt.       Task "Run Implementation" successful.         Generated logfile:       E:\TUpjWORKSPACE\R23b\_AASC\code\       14

### FPGA Area and Speed Optimization with MBD - Example

**Example:** Motor Speed Calculation requires the arithmetic operation "**Reciprocal**" which consumes lots of FPGA LUT resources, also challenging to meet timing constraints.



MathWorks AUTOMOTIVE CONFERENCE 2024

### FPGA Area and Speed Optimization with MBD – LUT Optimization I

Use the App "Lookup Table Optimizer" to re-target the design from FPGA LUT into RAM Blocks, reducing the use of LUT resources and improving timing performance.

#### (1) Open the lookup table optimizer app

	DEBUG	MODELING	FORMAT	ADDS	HDI CODE	SUBSYSTEM BLOCK		
SIMOLATION	DEDUG	MODELING	TORMAT	Arro	TIDE CODE	SOBSTSTEW BLOCK		
	Search							
Get	TAVORITES							
Ons + ONMENT	Linearization Manager	Model Linearizer	Control System Designer	Embedded Coder	Fixed-Point Tool	Requirements Manager	Coverage Analyzer	Simulink Test
	SIMSCAPE							
	Variable Scaling Analyzer	Load-Flow Analyzer						
	CONTROL SYSTEMS							
	Steady State Manager	Linearization Manager	Model Linearizer	Frequency Response	Control System Designer	Control System Tuner	Model Discretizer	
	SIGNAL PROCESSING	G AND WIRELESS C	OMMUNICATIONS					
	Logic Analyzer							
	CODE GENERATION							
	Embedded Coder	Simulink Coder	HDL HDL Coder	Fixed-Point Tool	Single Precision Converter	Lookup Table Optimizer		

#### (2) Choose MATLAB function handle

Objective	>	Setup
Welcome to th	e Lookup Table (LUT) (	Optimizer
Select the source for memo	ry efficient LUT	
O Simulink block or subsys	stem	
Create a memory efficient LU	JT for a Simulink block	
MATLAB Function Hand	le	
Create a memory efficient LU	JT for a MATLAB function har	ndle
O Fitted Curve		
Create a memory efficient LU	JT for a curve fit object from b	oase workspace

MathWorks **AUTOMOTIVE CONFERENCE 2024** 

# FPGA Area and Speed Optimization with MBD – LUT Optimization II

Specify function, enter the targeted input data type and input range, set to HDL Optimized

nspector - untitled" – 6	×		
AASC_RPM_2_RT			
0.0			
0.0			
0.001	Lookup Table Optimizer	7	
0.0001	Objective > Setup > Create	8	
RPM_Target III Estimated_Moto_Speed_RPM	MATLAR Function Handle		table, specified as a scalar.
000RPM			Relative tolerance measures
	@(x) 1/x	LUT Specification	×
	- Attributes of Memory Efficient   UT	Interpolation :	
1000		Progknaint enacification :	
ados sobre og 0000159 sobre odre ados ados ados ados ados 3 000097 pa ados ados ados ados ados ados ados ado		Dieakpoint specification .	ExplicitValues -
100RPf	Desired Out ut Data Type fixdt(0,16)	Saturate to output type :	False
	Inpu Desired Data Type Minimum Maximum	AUTOSAR Compliant :	False 🔹
	1 fixdt(0,29,27) 0.001 0.01	Explore Half :	True 💌
		HDL Optimized :	True 💌
		Solution Type :	Simulink •
[02] andice cadore cod (0000133) cadore codore codore codore codore (0000007) so codore codore codore codore codore 0000442 pod	0.00048		
Test Assessment/_itstop_t_tverify(input1 < 1) Test Assessment/_itstop_t_2verify(input1 < 1) Test Assessment/_itstop_t_3verify(input1 < 1)			
		Help	Ok Cancel
			permany second second second
		ι. Έ	o spend looking for a solution, by

### FPGA Area and Speed Optimization with MBD – LUT Optimization III

Tool generates new model targeting HDL efficiently



- CLB LUTs reduced from 887 to 167
- Achieved Timing closure!

IMPLEMENTED DE SIGN - xcau10p-sbvb484-1-i								
Tcl Console Messages Log Reports	Design Runs Ut	Itilization × Timi	ng Power I	lethodology DRC	Package Pins	I/O Ports		
Q, ₹ ≑	Q	% Hierarchy	here a real of the					
Hierarchy Summary	Name	^ CLB LUTs (44000)	CLB Registers (88000)	CARRY8 F7 Muxes (9720) (38880)	s F8 Muxes (19440)	CLB LUT as (9720) (440	s Logic Block RAM 000) Tile (100)	DSPs (400)
✓ CLB Logic	N RPM_2_RT_C	Calc 167	128	14 1	9 4	43	167 0.5	5
Tcl Console     Messages     Log     Reports     Des       Q     ★     ♦     C     ₩     ●	ign Runs Utilization	on Timing x Po Summary	ower Methodolo	gy DRC Package F	Pins I/O Ports			
General Information Timer Settings	Setup		Hold			Pulse Width		
Design Timing Summary	Worst Neg	gative Slack (WNS): 0	0,719 ns Wo	orst Hold Slack (WHS):	0,056 ns	Worst Pulse Wid	th Slack (WPWS):	1,457 ns
Clock Summary (1)	Total Neg	gative Slack (TNS):	0,000 ns Tot	tal Hold Slack (THS):	0,000 ns	Total Pulse Width	h Negative Slack (TPWS):	0,000 ns
Methodology Summary (52)	Number o	of Failing Endpoints: 0	0 Nu	mber of Failing Endpoint:	s: 0	Number of Failin	g Endpoints:	0
Check Timing (47)     Dutra Clock Baths	Total Num	mber of Endpoints:	316 Tot	tal Number of Endpoints:	316	Total Number of I	Endpoints:	133
Inter Clock Paths	All user speci	ified timing constraint	ts are met.					

MathWorks AUTOMOTIVE CONFERENCE 2024

### FPGA Area and Speed Optimization with MBD

- Simulink Test provides non intrusive test harnesses for model tests and efficient integrated workflows.
- Simulink Coverage provides Model Coverage
- HDL Verifier proves the equivalence between the model and generated code.
- HDL Coder is also certified by TÜV SÜD to be suitable for use in developing ISO 26262 products for all ASILs when taking V&V measures







21

### FPGA Verification with MBD – Model Verification

Use **Simulink Test** to create a test harness, to evaluate input / output data range, data types, testbench to define expected reference behaviors.



### FPGA Verification with MBD – Measuring Test Coverage

Achieve 100% Coverage with the same Testbench against the new, efficient model (MiL).

Test Harness: RPM_2_RT_Calc_Harness/RPM_2_RT_Calc * - Simulink				
SIMULATION DEBUG MODELING FORMAT	APPS COVERAGE X	HARNESS		
Coverage ON STATLIS CON CON Collection Collection Collection Collection Collection Contend Con	Analyze Step Stop overage - Forward	Coverage Results Ge Details Explorer P REVIEW RESULTS	enerate Report	
Image: Apple of the second	Coverage Det	ails		
•	😮 Detail	S		
	1. SubS	vstem block " <u>RPM_2_</u>	RT_Calc"	
⇒	Chil	d Systems: Fraction	BitSlice1, IndexBitSlice1	
	Met	ric	Coverage (this object)	Coverage (inc. descendants)
	Exe	rution	NA	100% (30/30) objective outcomes
	Switch	block " <u>SaturateUpperBou</u>	undl"	
	Justi	fy or Exclude		
	Pare	nt: <u>RPM_2</u>	RT_Calc_Harness/RPM_2_F	<u>RT_Calc</u>
	Met	ric	Coverage	
	Exe	rution	100% (1/1) objective outcomes	
	Exe	cution analyzed		
	BI	ock executed		100%
				875001/875001

### FPGA Verification with MBD – Back-to-Back Verification

(7) **HDL Verifier** validates that the model and generated HDL code have the same behavior as in HDL simulator (co-simulation) and on FPGA Hardware (FPGA-in-the-Loop). It can also incorporate legacy hand code into Simulink models for simulation. Compare model output

Open cosimWizard 

Note: The second	×		
Steps	Actions		
-> Cosimulation Type HDL Files HDL Compilation	Select the type of cosimulation you want to do. If the HDL simulator executable you want to use is not on the system path in your environment, you must specify its location.		
Simulation Options Input/Output Ports Output Port Details Clock/Reset Details Start Time Alignment Block Generation	HDL cosimulation with: Simulink   HDL Simulator: ModelSim		
	<ul> <li>Use HDL simulator executables on the system path</li> <li>Use the HDL simulator executables at the following location</li> </ul>		

Specify source code, sample time, data types, reset timing... 

Steps	Actions					
Cosimulation Type -> HDL Files HDL Compilation Simulation Options	Add all VHDL, Verilog, and/or script files to be used in cosimulation to the following table. If the file type cannot be automatically detected or the detection result is incorrect, specify the correct file type in the table. If possible, we will determine the compilation order automatically using HDL simulator provided functionality. Then the HDL files can be add					
Input/Output Ports	mes and data types to 'Inherit'				HDL tim	
Input/Output Ports	Source riles.	List				
Output Port Details		Sample Time	Data Type	Sign	Clocks	
Clock/Reset Details	E:\TU\prj2111\ScanNumbe	1	Inherit •	Signed		
Start Time Alignment	E:\TU\prj2111\ScanNumbe	4e-9	Inherit •	Signed	Clock	
Block Generation	E:\TU\pri2111\ScanNumbe	4e-9	Inherit .	Signed		
		4e-9	Inherit •	Signed	CIK	
	[E:\TU\prj2111\ScanNumbe	o 4e-9	Inherit •	Signed		
		o 4e-9	Inherit .	Signed	- Inherit	

against HDL simulator output HDL simulator 24 9.96.9 sa16 truth... Arch scannum\_s. scannum s. scannum s. time unit ins 10 15 20 ck Name Period(ns) Active Edge Rising eset 10 15 20

### Key Takeaways

Aspects	<b>Project: Scala 2</b> Bottom-up, Manual Approach	Project: Scala 3 MBD Approach
Executable specification	Requirements and design implementations are separate files. High efforts in understanding the design and maintaining consistency and traceability.	Model serves as both simulatable requirements and design implementation. Easy to understand, high confidence level in consistency throughout V-Cycle. Automatic traceability.
System-level simulation	Surprises can occur late in the design process. Can be challenging to fix. Different teams use diverse tools. Domain knowledge not as easy to share.	Cross-domain teams collaborate in the same Simulink environment to investigate ideas, possibilities, limits, and identify risk early.
What-if analysis	Limited options explored as HW implementation. Not easy, high cost to perform generic thorough investigation.	Many design options, scenarios and trade study can be simulated and evaluated early on HW
Automatic production code generation	Hand coding is prone to manual errors, experience and time consuming. Requirement fulfillment can be difficult to understand.	Systematic, automatic approach. Higher efficiency.
Knowledge management	High effort, difficulties in communication, and in convincing different teams and stakeholders.	Teams / stakeholders communication easier

#### Gratitude to MathWorks Professionals and Consultants

"MathWorks Experts collaborated with Valeo throughout project development, offering expertise in Model-Based Design, guidance, issue investigation, technical evaluation and design optimization - from the early design phase to project completion."

#### Thank you for supporting Valeo's project success !!

#### MathWorks **AUTOMOTIVE CONFERENCE 2024** Europe

# Thank you

