MATLAB EXPO 2018

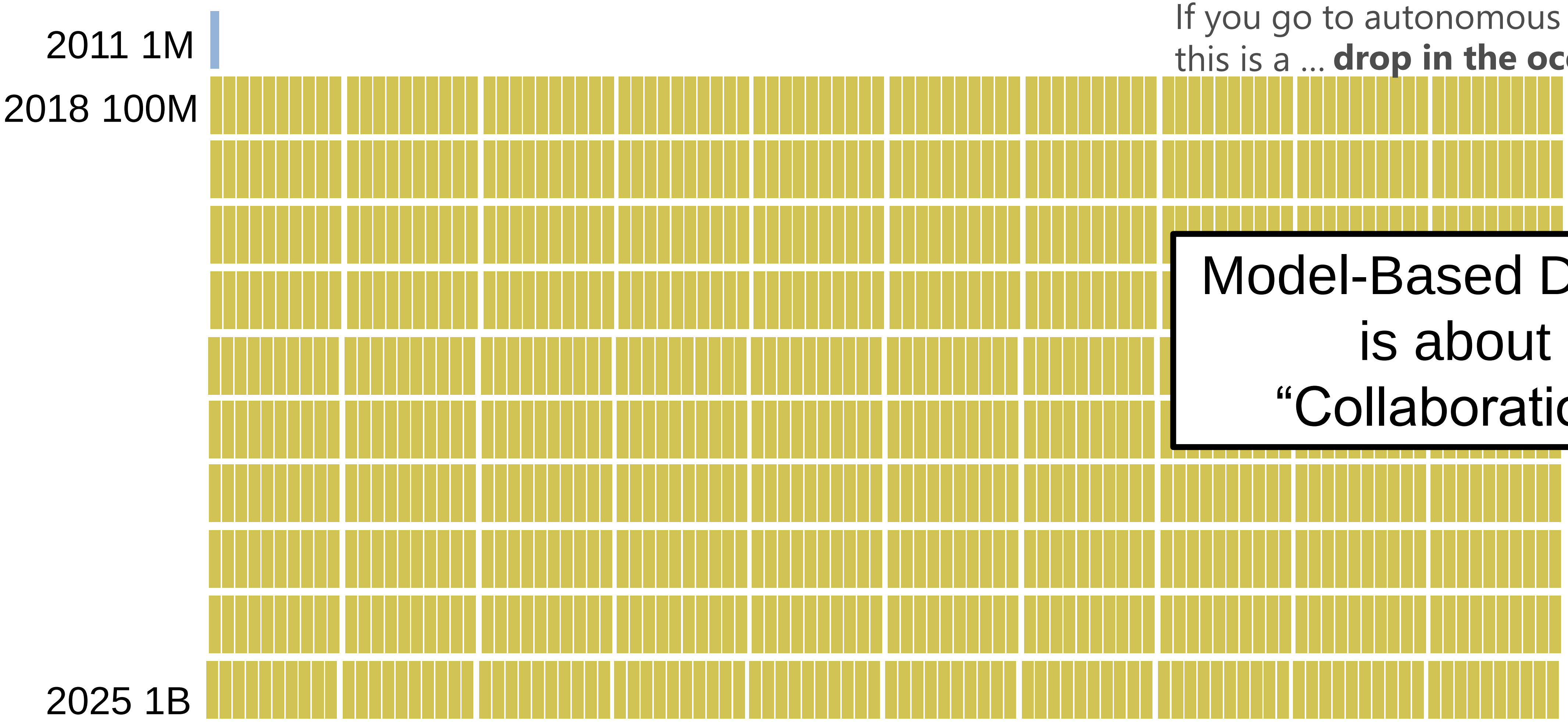
Generating Industry Standards Production C Code Using Embedded Coder

Rajat Arora
Durvesh Kulkarni

Lines of Code (LOC) is exploding

"We've got 100 million lines of code in the 7 Series.

If you go to autonomous driving this is a ... **drop in the ocean**."



Model-Based Design is about "Collaboration"

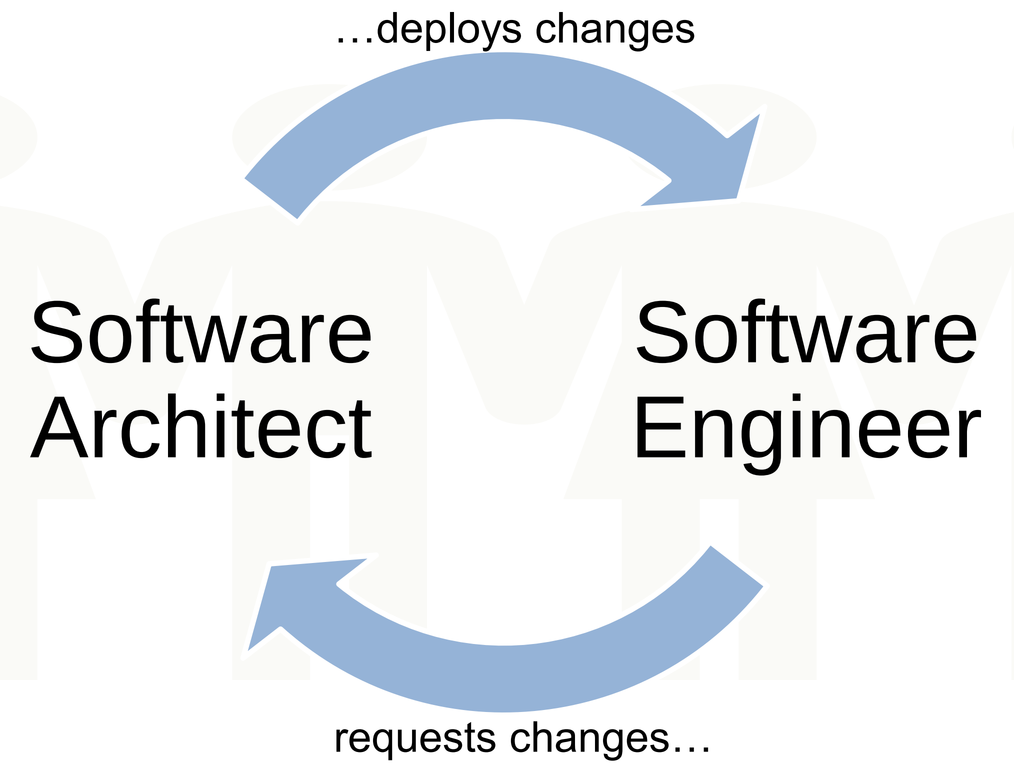
www.motoring.com.au/autonomous-cars-may-never-go-global-says-bmw-106430/

Join Hands & Develop

*…"standardization by **customizing** the Simulink® development environment"…*

A software architect helps
- multiple engineers
- multiple projects

generate code that ***conforms*** to the organization standard



A software engineer directly ***interacts*** with and generates code from production models.

Join Hands & Develop

Software Architect

Software Engineer

…Collaborative Workflow…

Quick Study

Standardize Code Architecture

Production Code Details

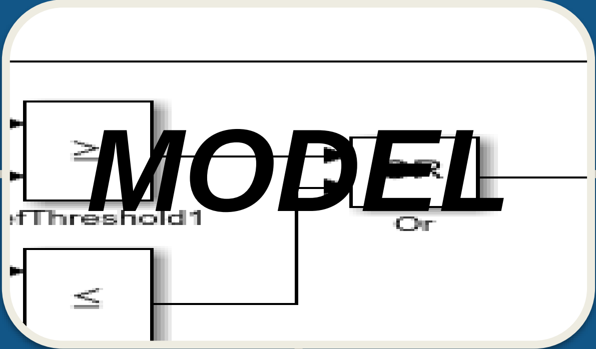
Improve Code Efficiency

Standards & Certification

Where Should I Start From ?

Compatibility

Configuration



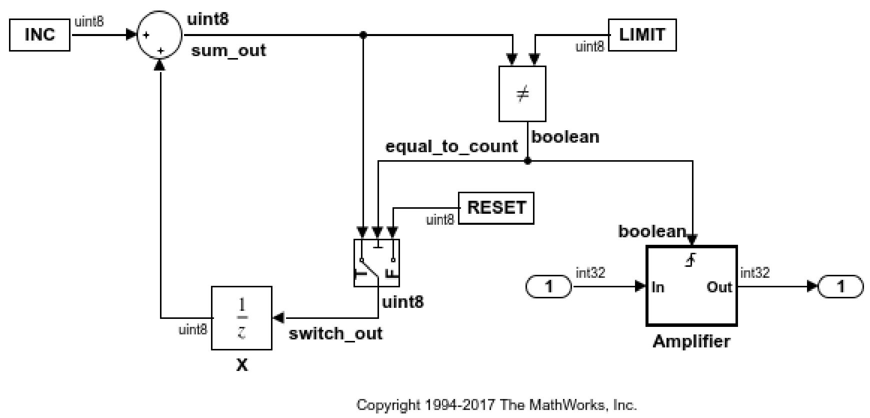
MODEL

Optimize

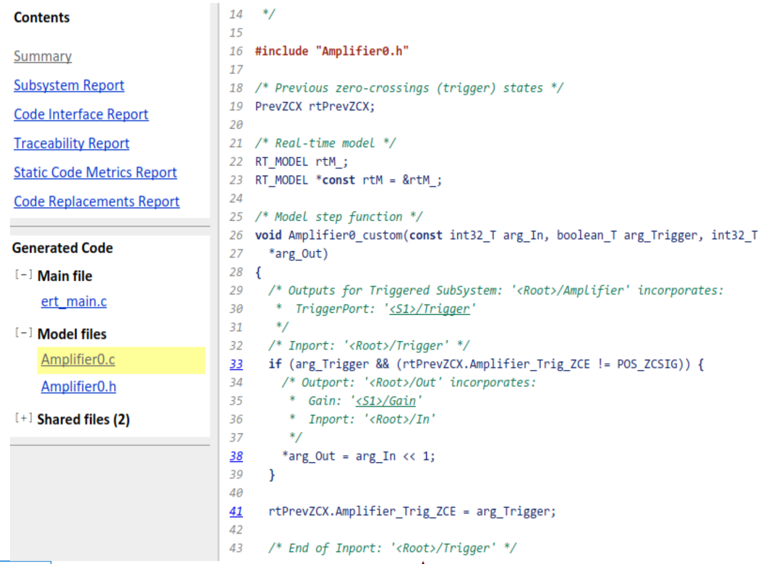
Interface

Grid Locked Situation !!!

Fast Code Generation Using Quick Start



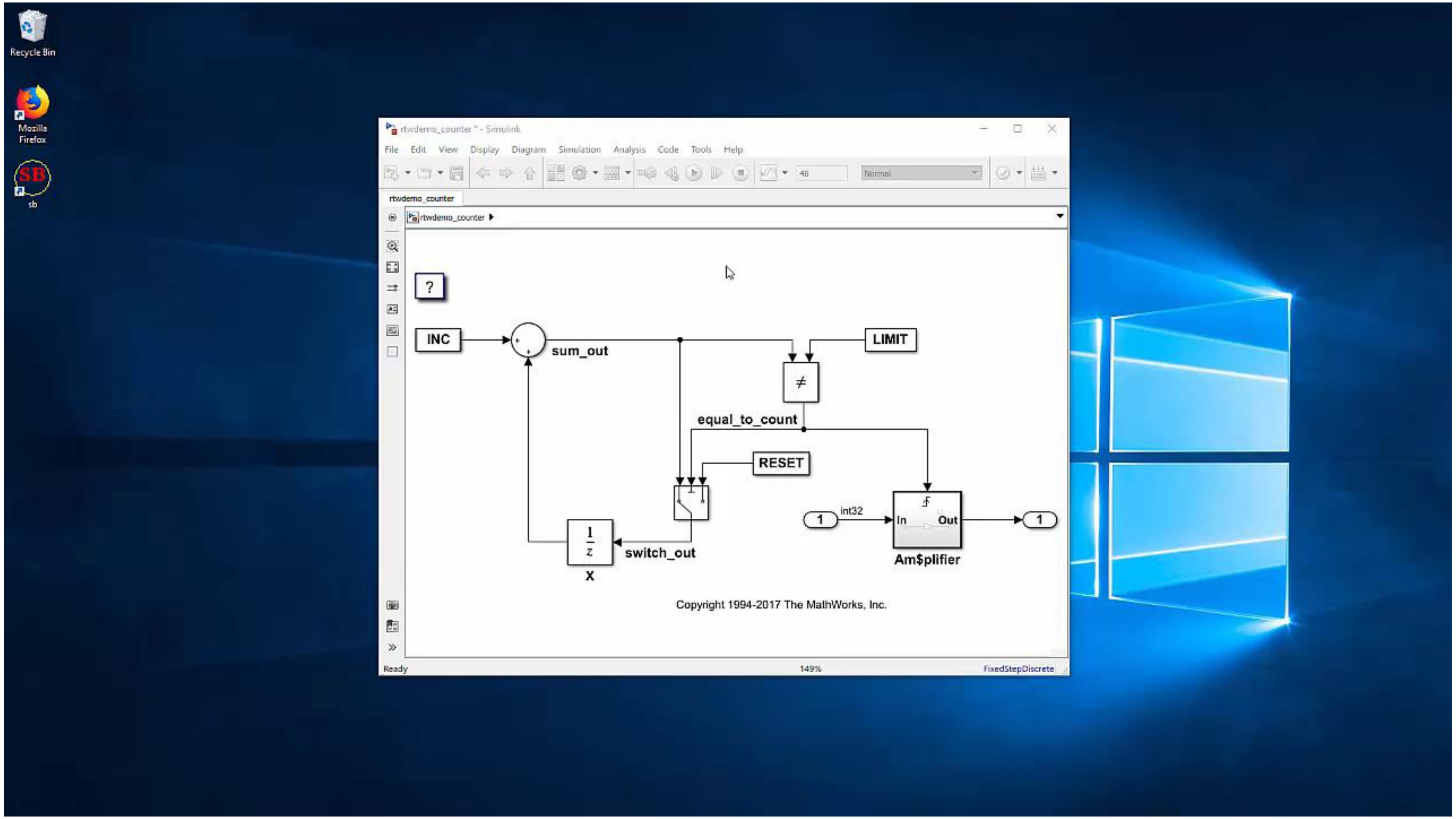
SIMULINK MODEL



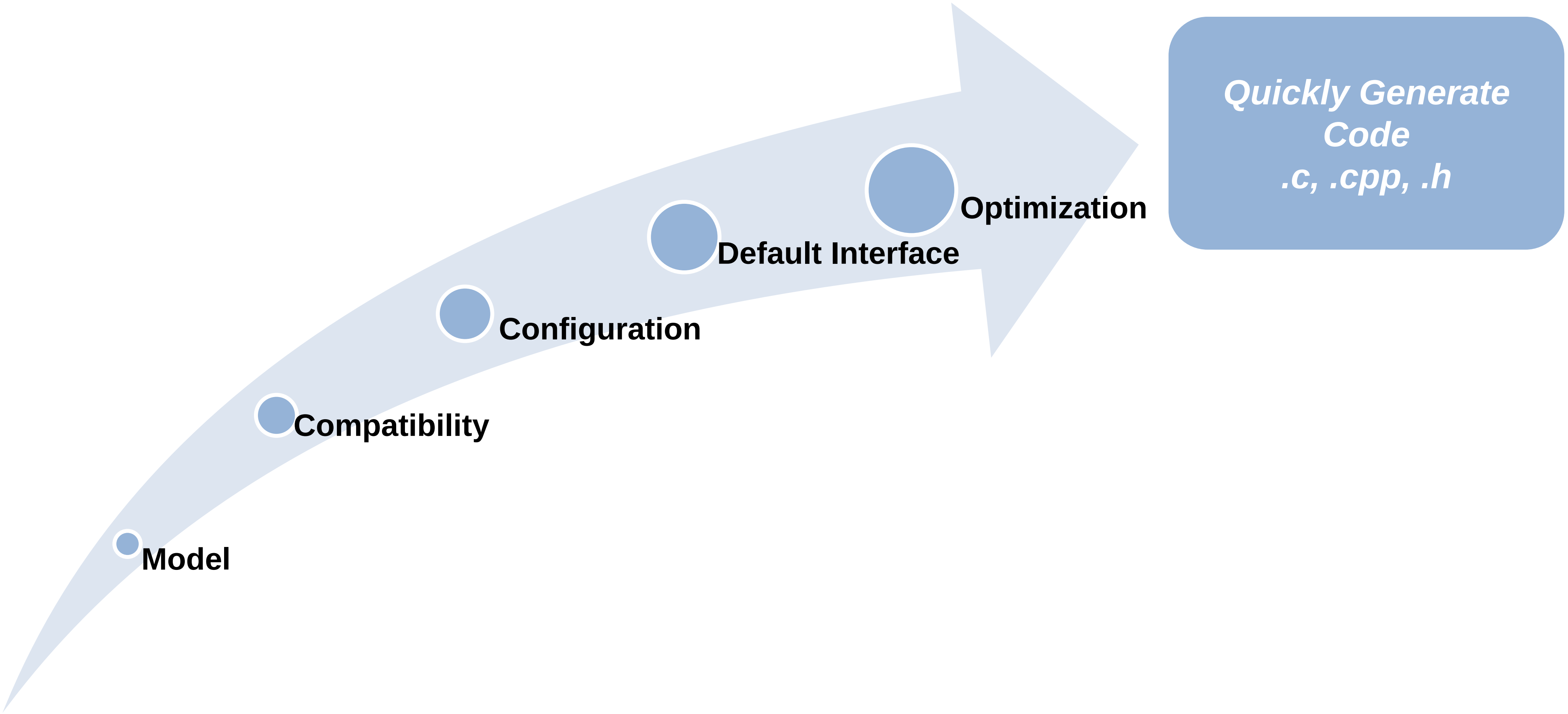
GENERATED CODE

QUICK START – 7 Simple Steps

7 Simple Steps for Generating Quick Code



Quick Start Wizard = True; Grid Solved = Quick Start Wizard;



Join Hands & Develop

System Architect

Software Engineer

…Collaborative Workflow…

Quick Study

Standardize Code Architecture

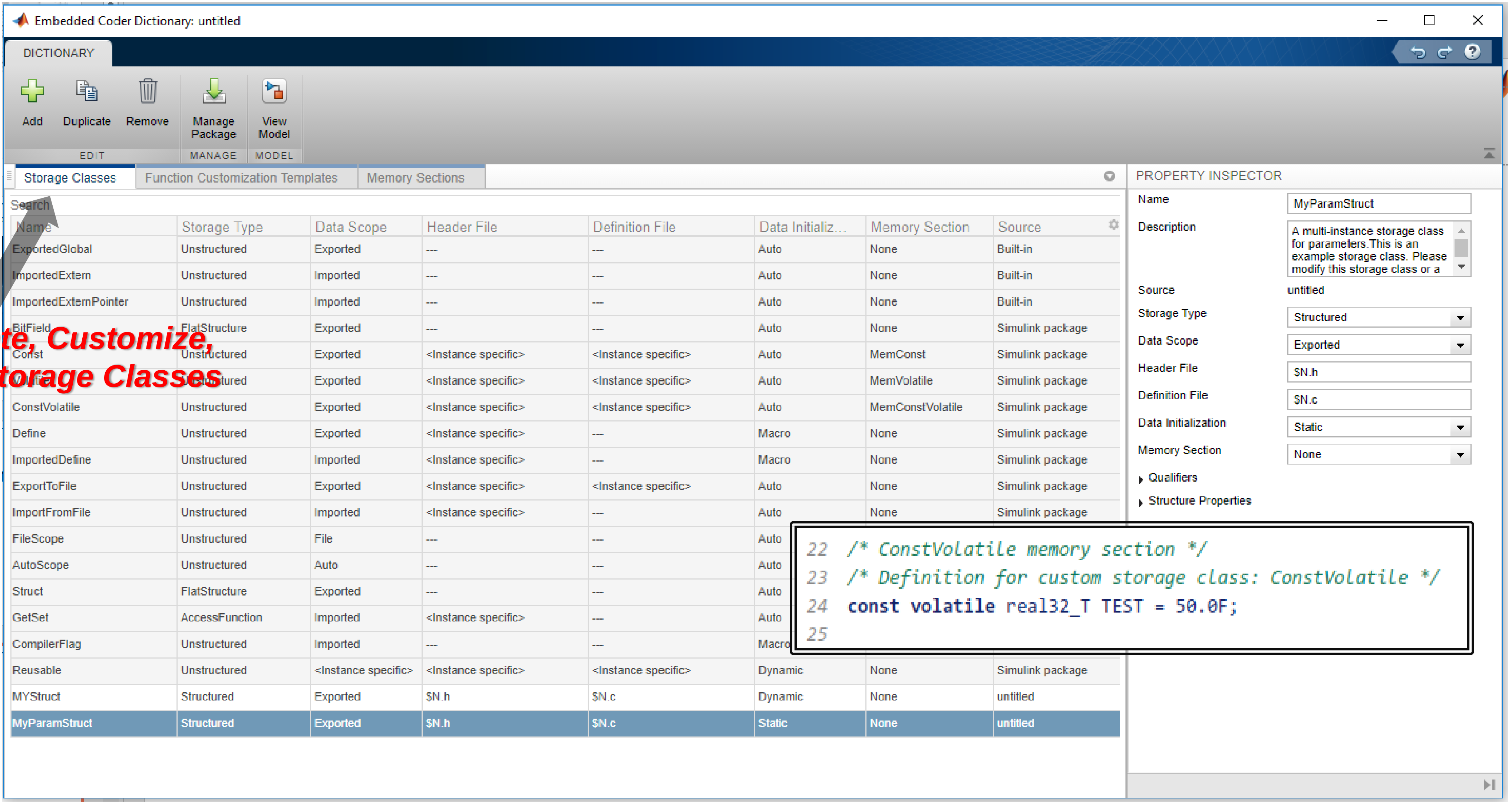
Production Code Details

Improve Code Efficiency

Standards & Certification

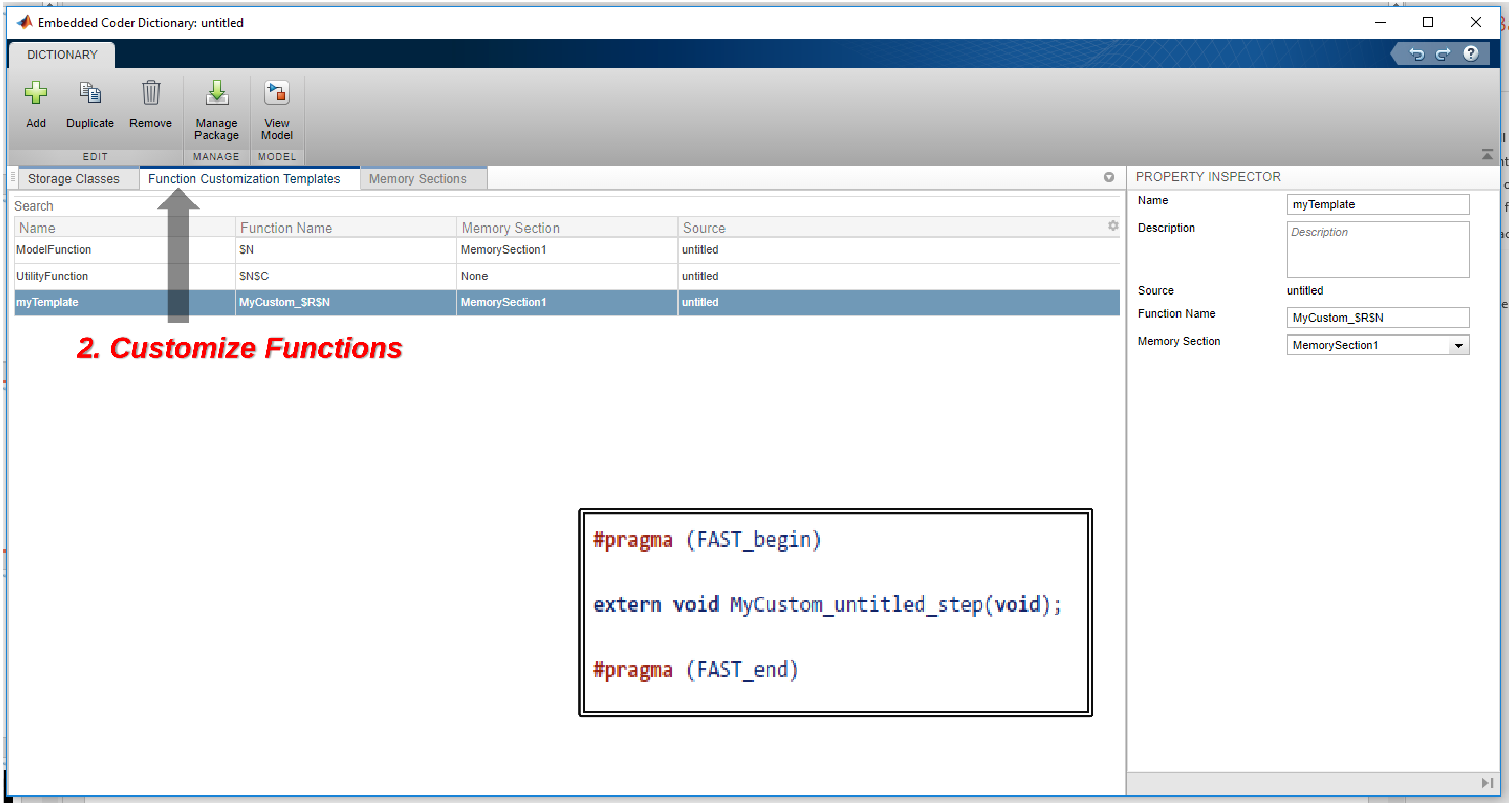
How to go about quickly architecting my code ?

The “Coder Dictionary” For Your Code Architecting Needs

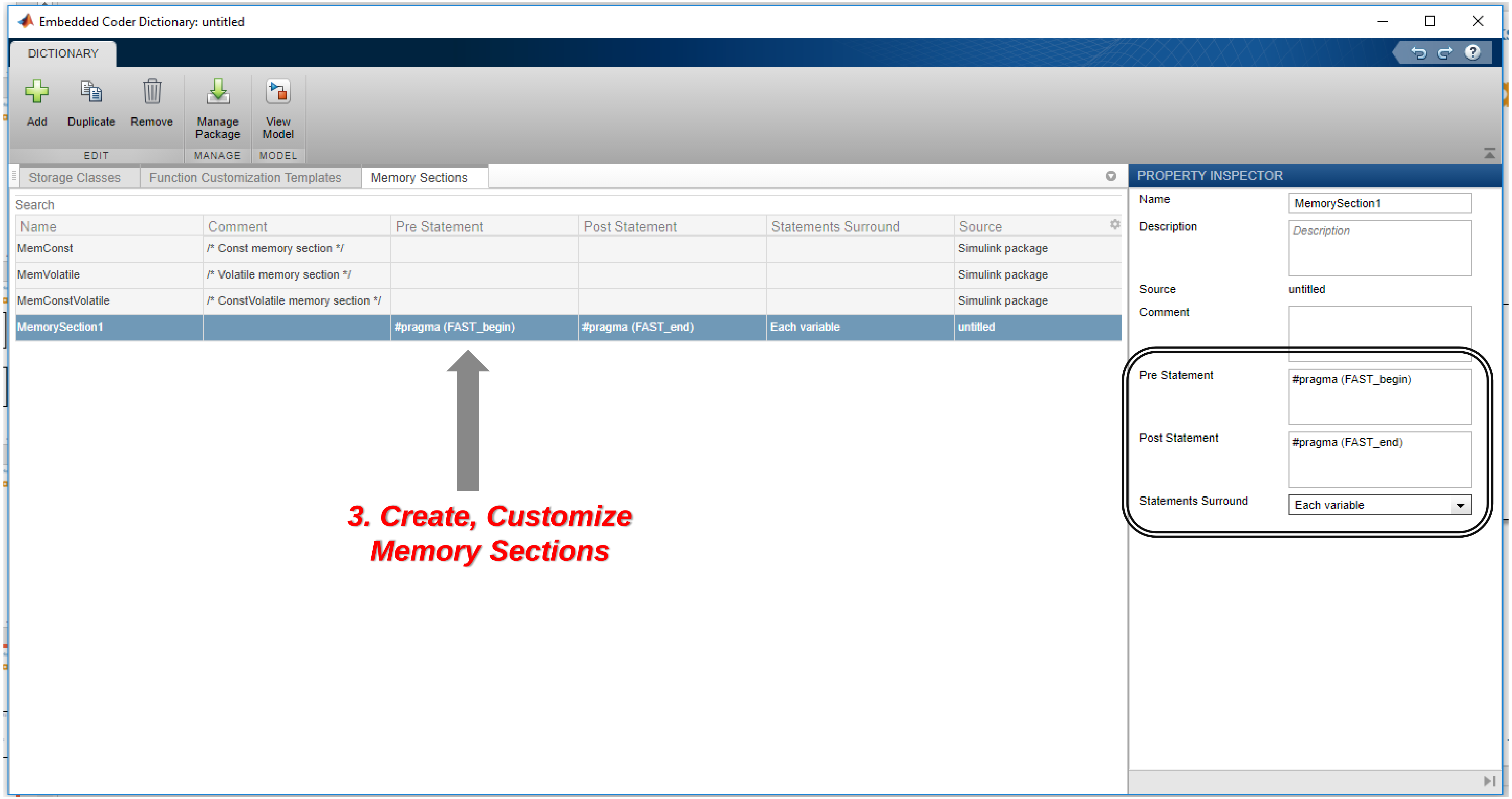


1.Create, Customize, Your Storage Classes

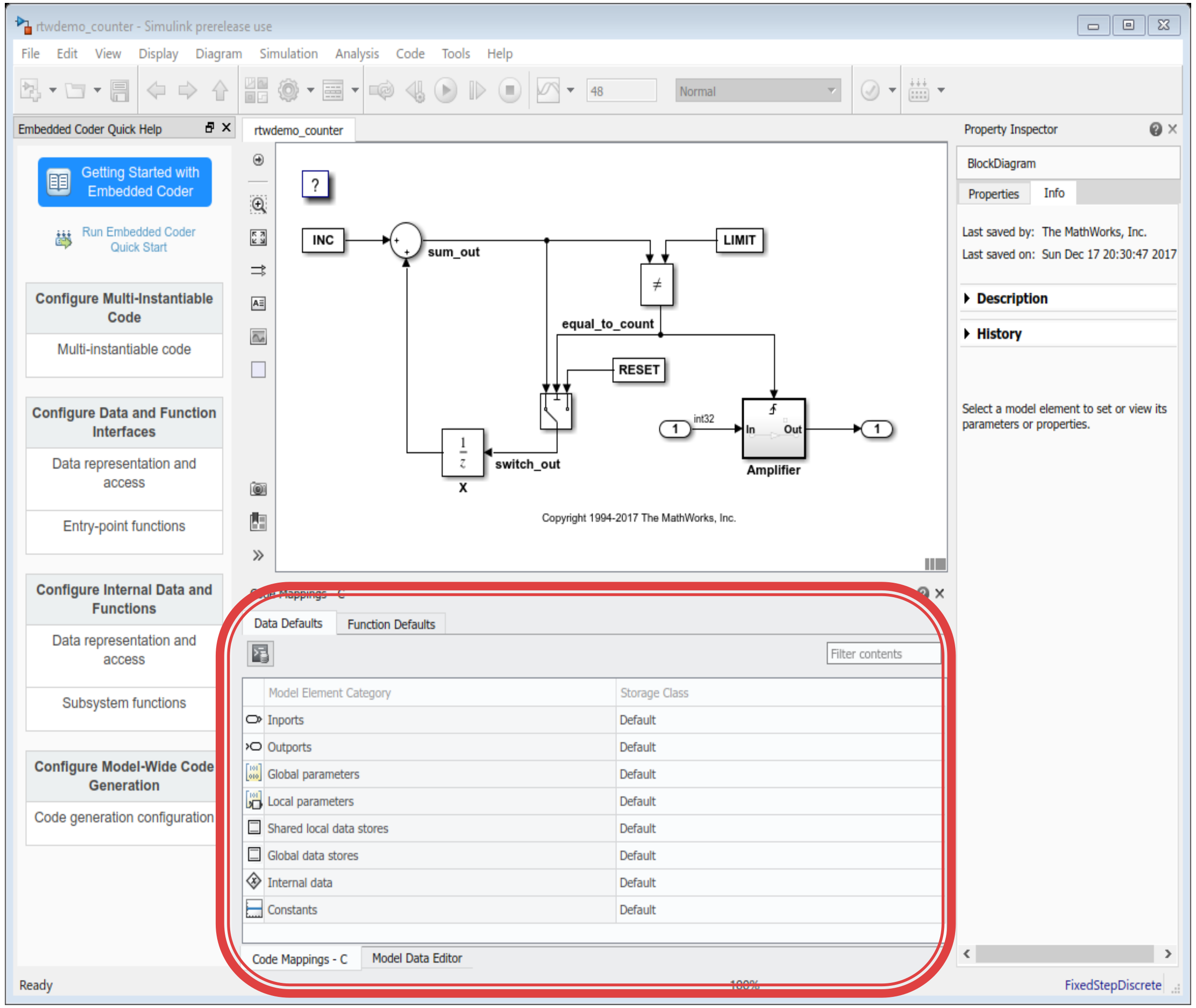
The “Coder Dictionary” For Your Code Architecting Needs



The "Coder Dictionary" For Your Code Architecting Needs



Easily configure your **Model** with Code Mapping Editor



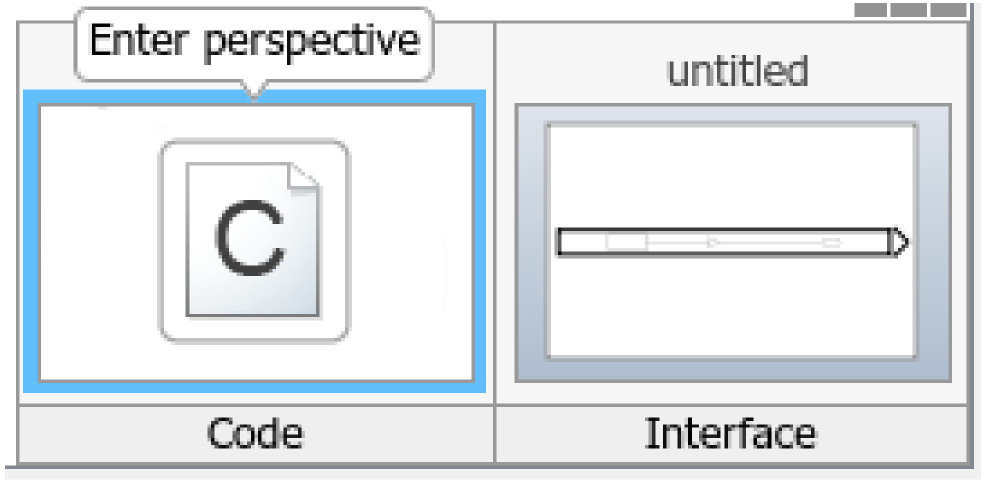

```
typedef struct {
  uint8_T X_Delay;
} DW;

typedef struct {
  ZCSigState Amplifier_Trig_ZCE;
} PrevZCX;

typedef struct {
  int32_T Input;
} ExtU;

typedef struct {
  int32_T Output;
} ExtY;
```

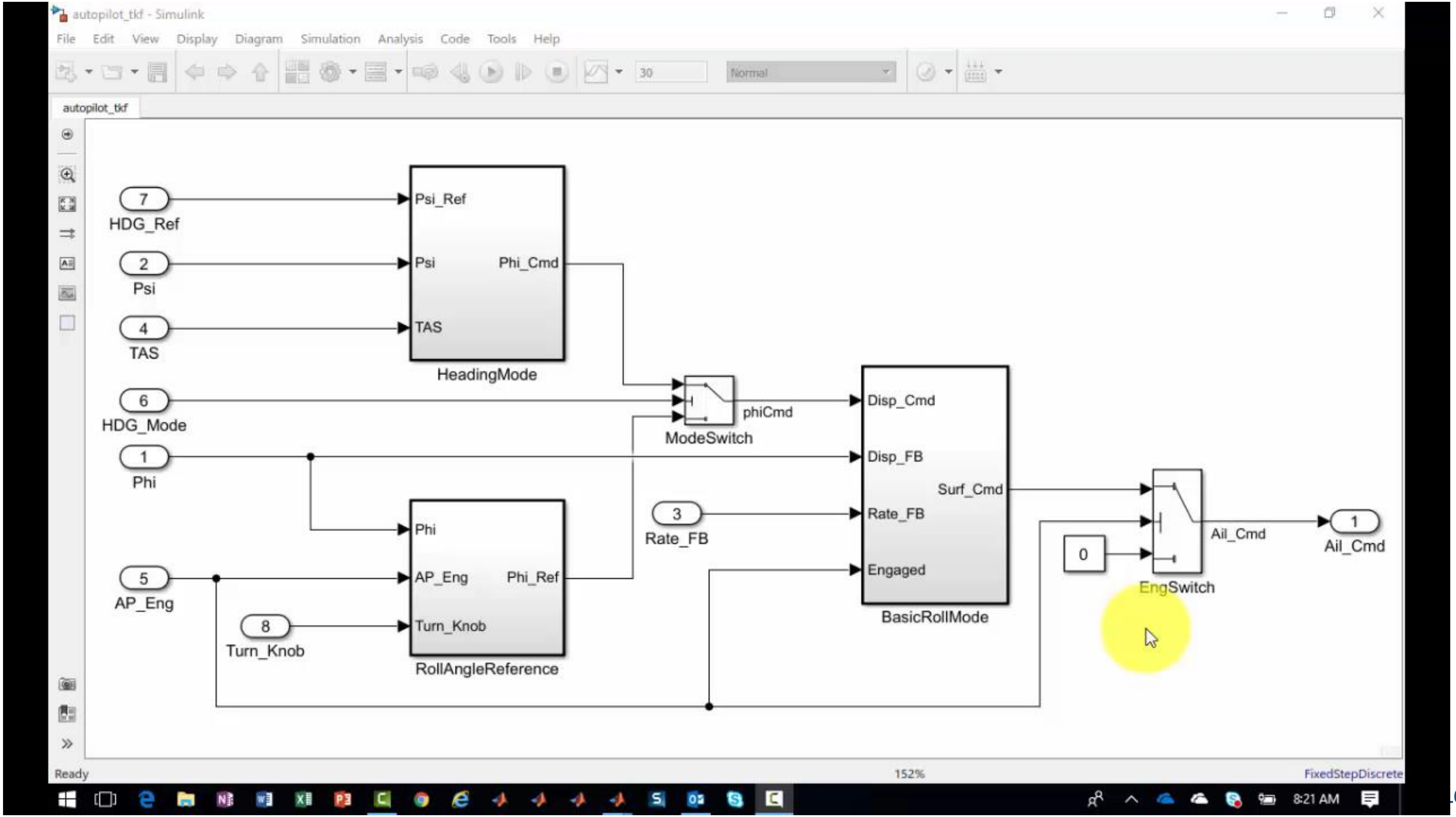
rtwdemo_counter.h

```
DW rtDW;
PrevZCX rtPrevZCX;
ExtU rtU;
ExtY rtY;
```

rtwdemo_counter.c

CoderPerspective
=
{'QuickStart','CoderDictionary','CodeMapping'}





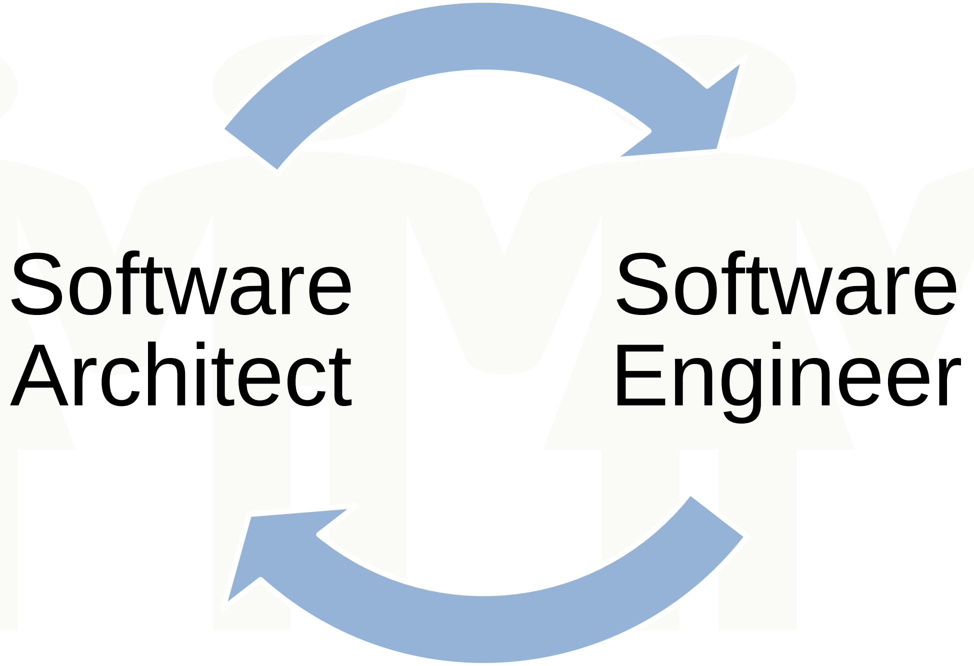
autopilot_tkf - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
30
Normal
autopilot_tkf
7
HDG_Ref
2
Psi
4
TAS
6
HDG_Mode
1
Phi
5
AP_Eng
8
Turn_Knob
Psi_Ref
Psi
TAS
Phi_Cmd
HeadingMode
Phi
AP_Eng
Turn_Knob
Phi_Ref
RollAngleReference
phiCmd
ModeSwitch
3
Rate_FB
Disp_Cmd
Disp_FB
Rate_FB
Engaged
Surf_Cmd
BasicRollMode
0
EngSwitch
Ail_Cmd
1
Ail_Cmd
Ready
152%
FixedStepDiscrete
8:21 AM

Join Hands & Develop

…"standardization *by **customizing** the Simulink® development environment"…*

Creates code definition for Data & Functions with:

- **Embedded Coder Dictionary**
- **Custom Storage Class Designer**



Configure default settings for model.

Apply Embedded Coder Dictionary code definitions with:

- **Code Mapping Editor**
- **Model Data Editor**
- **Function Prototype Control**

Join Hands & Develop

Software Architect

Software Engineer

...Collaborative Workflow...

Quick Study

Standardize Code Architecture

Production Code Details

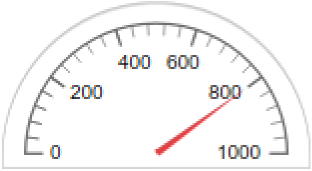
Improve Code Efficiency

Standards & Certification

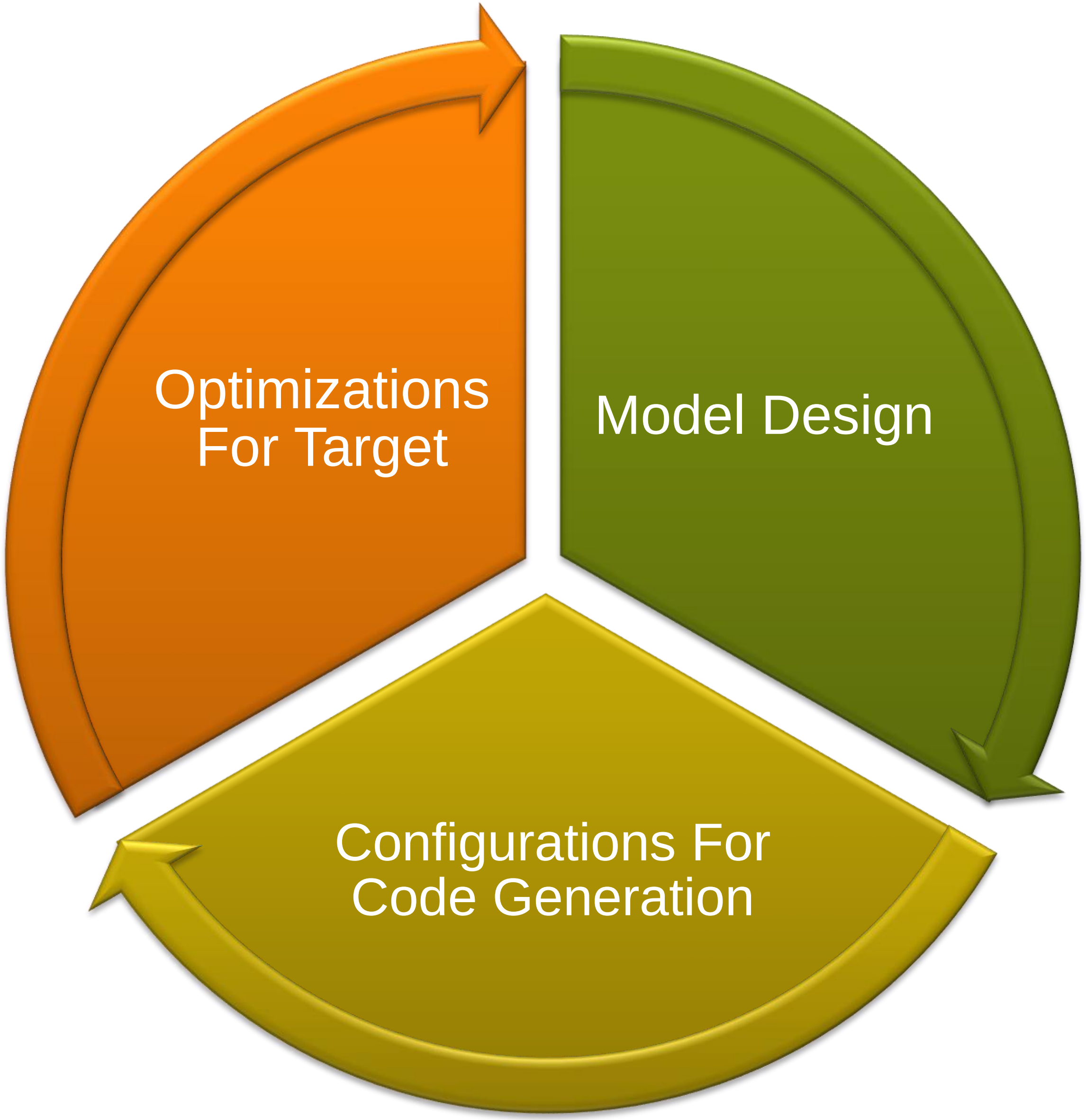
How and What Should I Optimize?

RAM

ROM

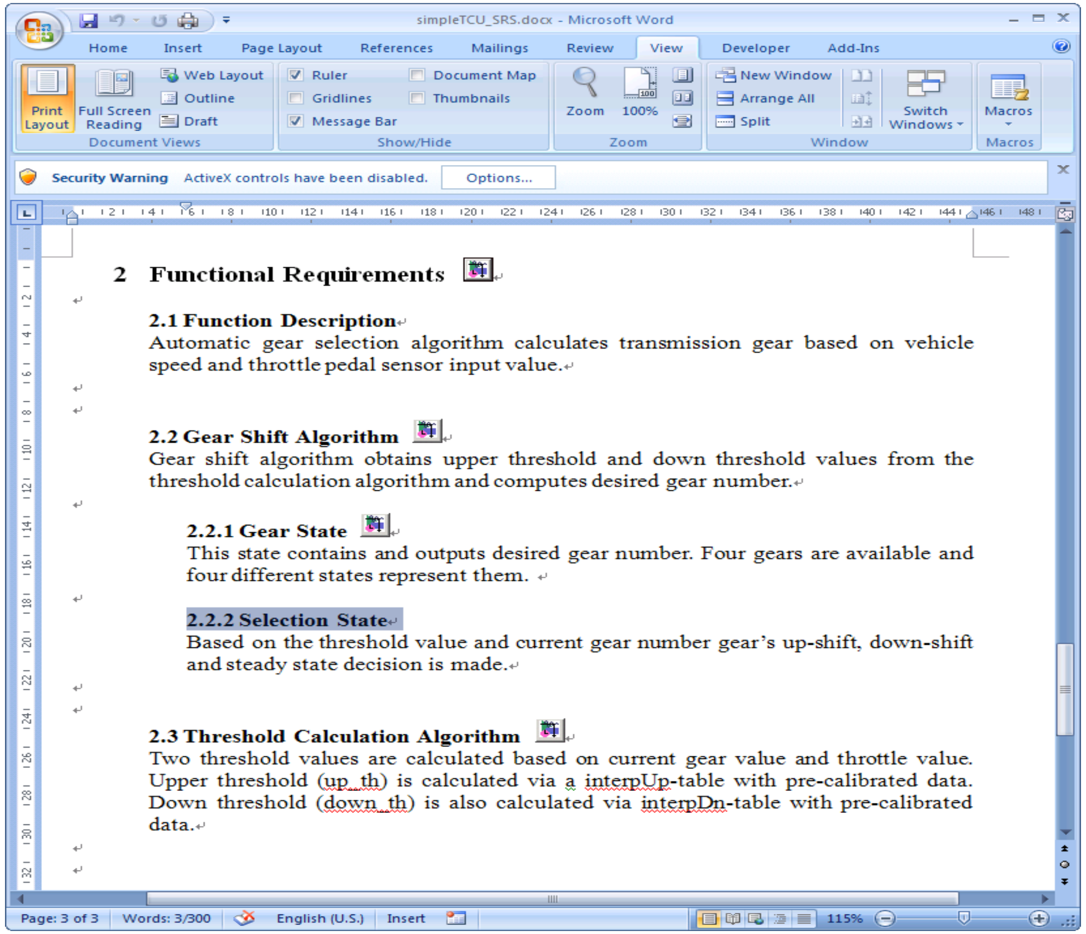


Execution Speed



Achieve Optimized Code Via Design

Design using requirements & simulate the model to observe behavior


simpleTCU_SRS.docx - Microsoft Word

2 Functional Requirements

2.1 Function Description
Automatic gear selection algorithm calculates transmission gear based on vehicle speed and throttle pedal sensor input value.

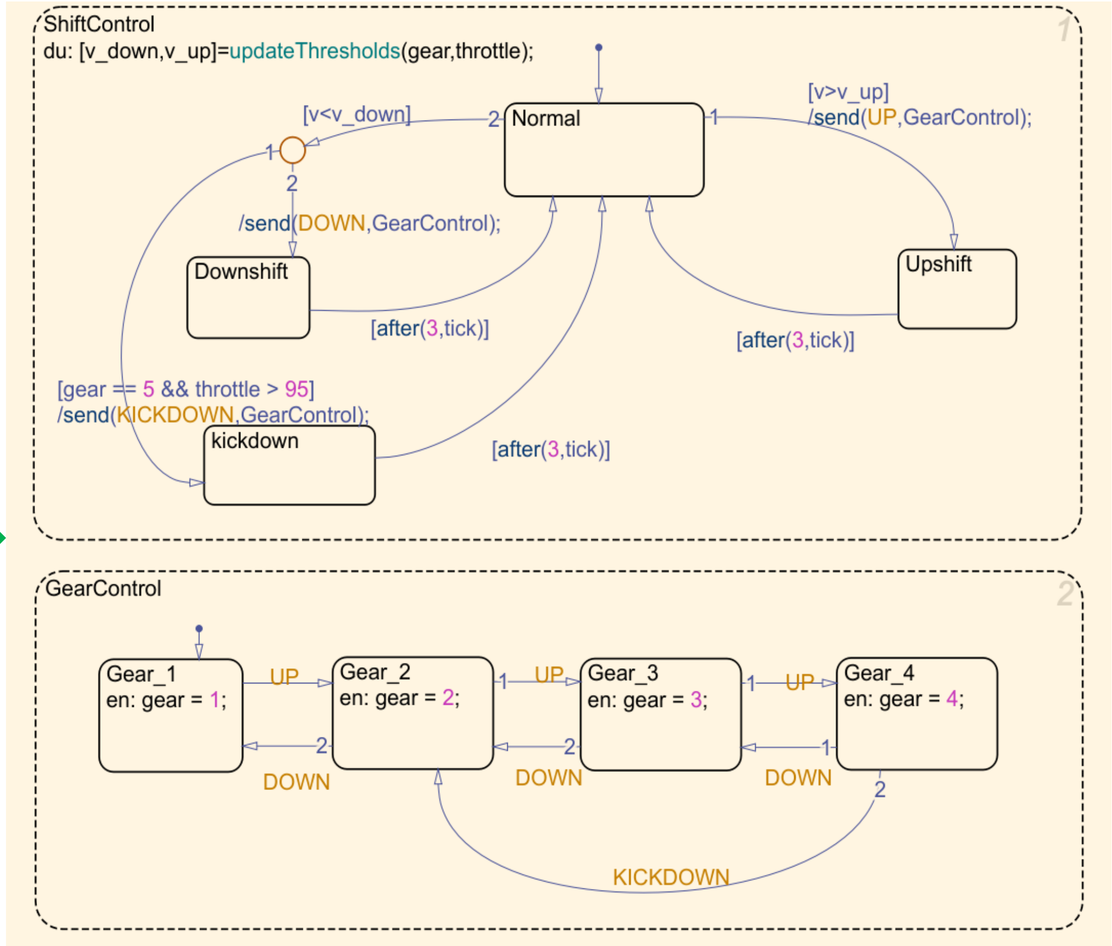
2.2 Gear Shift Algorithm
Gear shift algorithm obtains upper threshold and down threshold values from the threshold calculation algorithm and computes desired gear number.

2.2.1 Gear State
This state contains and outputs desired gear number. Four gears are available and four different states represent them.

2.2.2 Selection State
Based on the threshold value and current gear number gear's up-shift, down-shift and steady state decision is made.

2.3 Threshold Calculation Algorithm
Two threshold values are calculated based on current gear value and throttle value. Upper threshold (up_th) is calculated via a interpUp-table with pre-calibrated data. Down threshold (down_th) is also calculated via interpDn-table with pre-calibrated data.

Page: 3 of 3 Words: 3/300 English (U.S.) Insert 115%

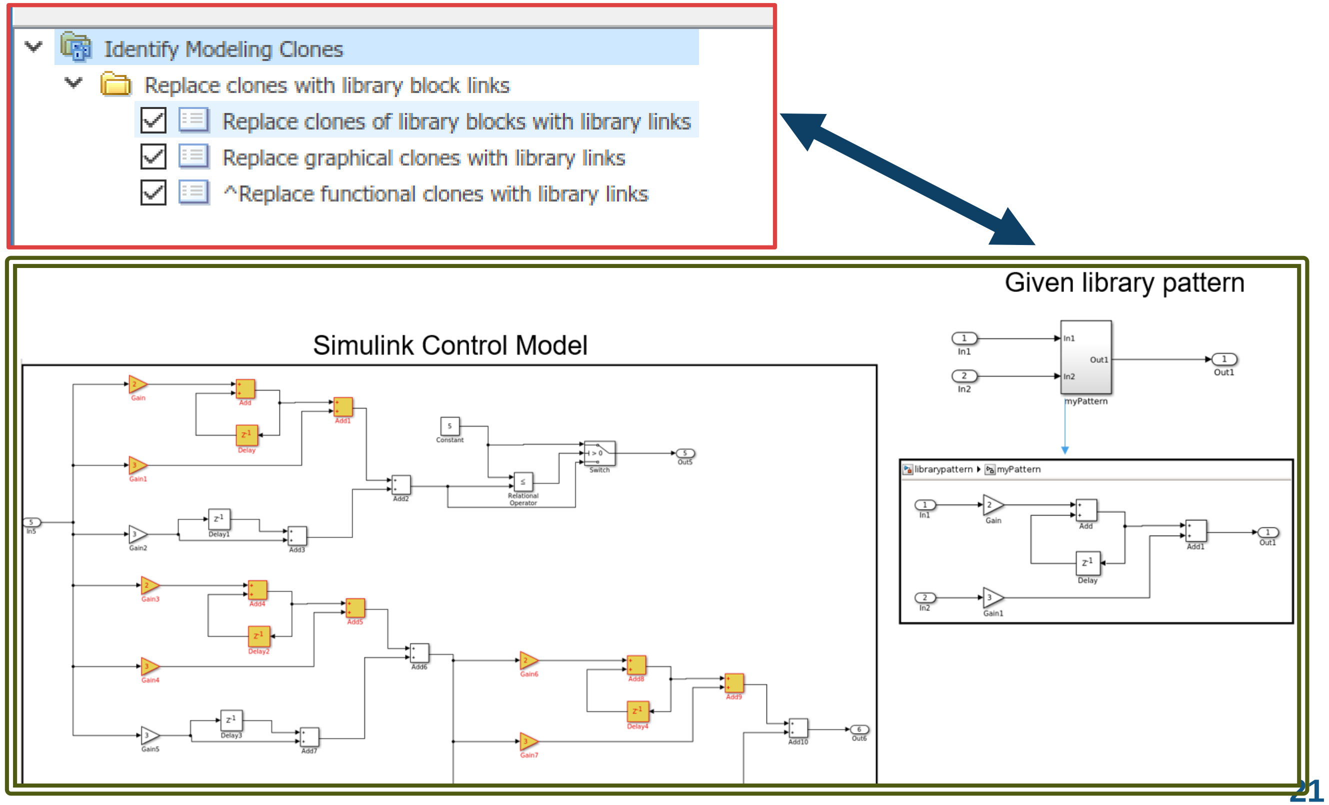


Achieve Optimized Code Via Design

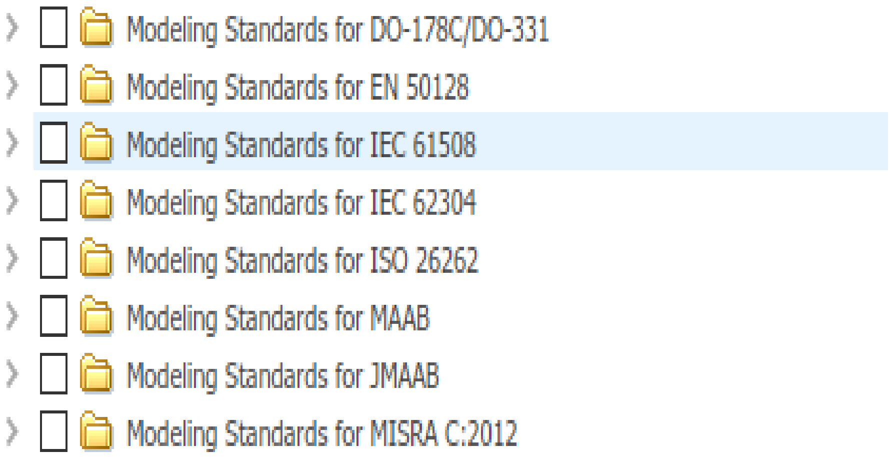
Use Model Advisor to apply and establish best Modeling Practices

- MAAB, MISRA, ISO/IEC/DO guidelines, etc.
- Simulink and Stateflow Guidelines
- Clone Detection

Clone Detection



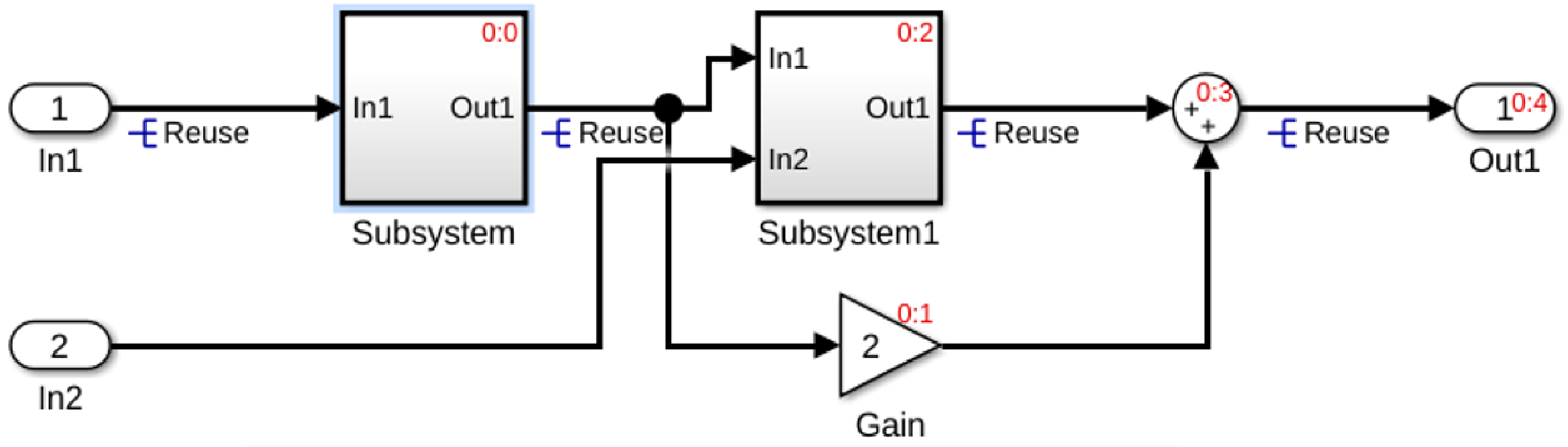
Model Advisor Guidelines



Achieve Optimized Code Via Design

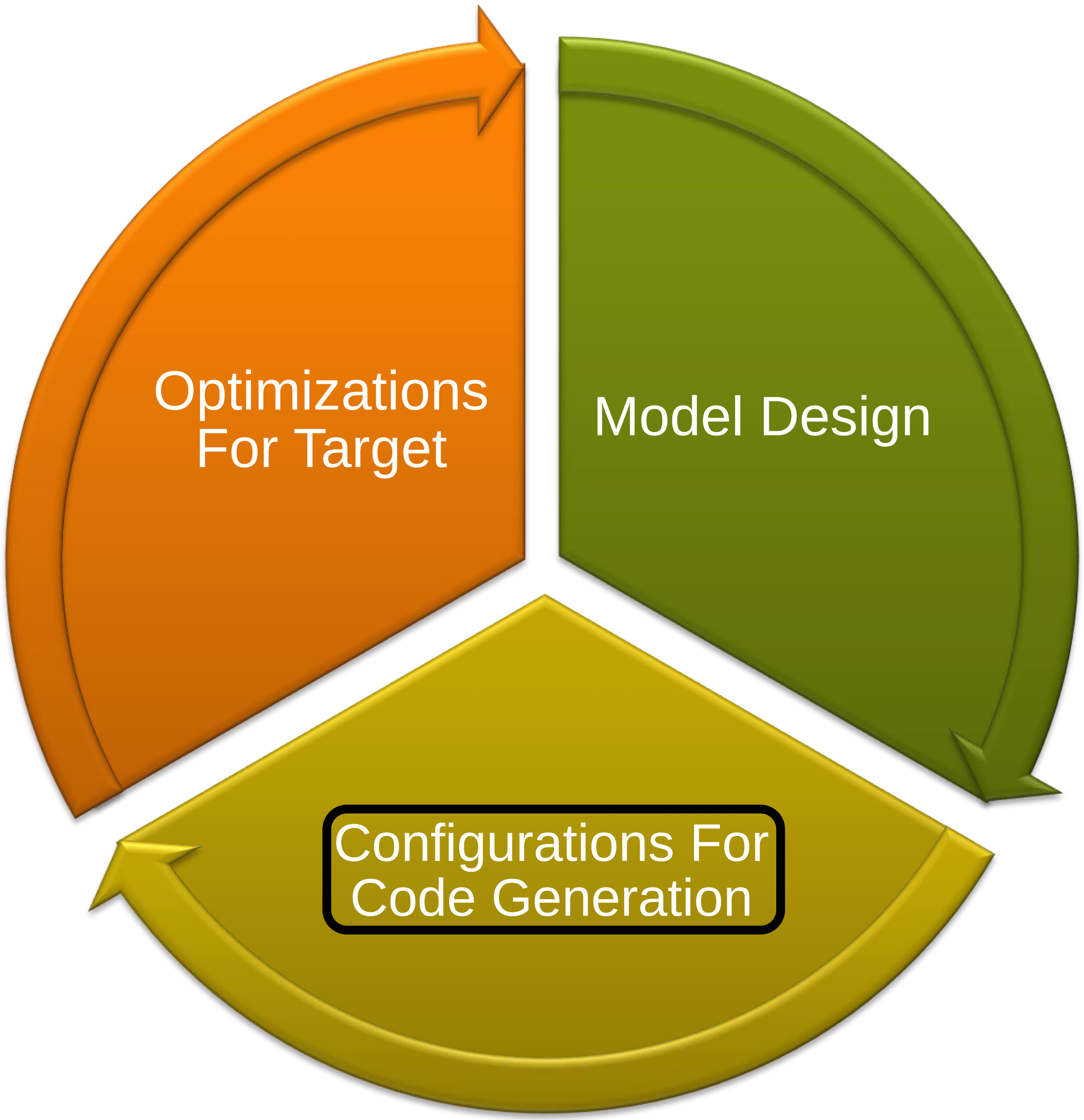
Use Simulink Data Objects and Custom Storage Classes to optimize

- E.g. Reusable (Data copy reduction), Localizable (Force local vs global use)

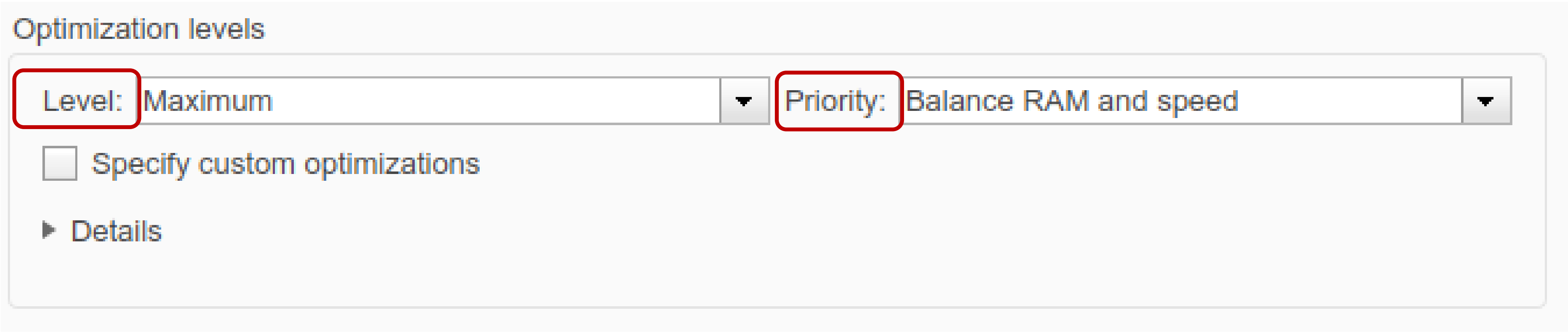



```
void Subsystem(void)
{   ...
    for (i = 0; i < 100; i++) {
        Reuse[i] = 2.0F * Reuse[i] * 20.0F;
    }
    fcn3(&Reuse[0]);
}
```

How and What Should I Optimize?



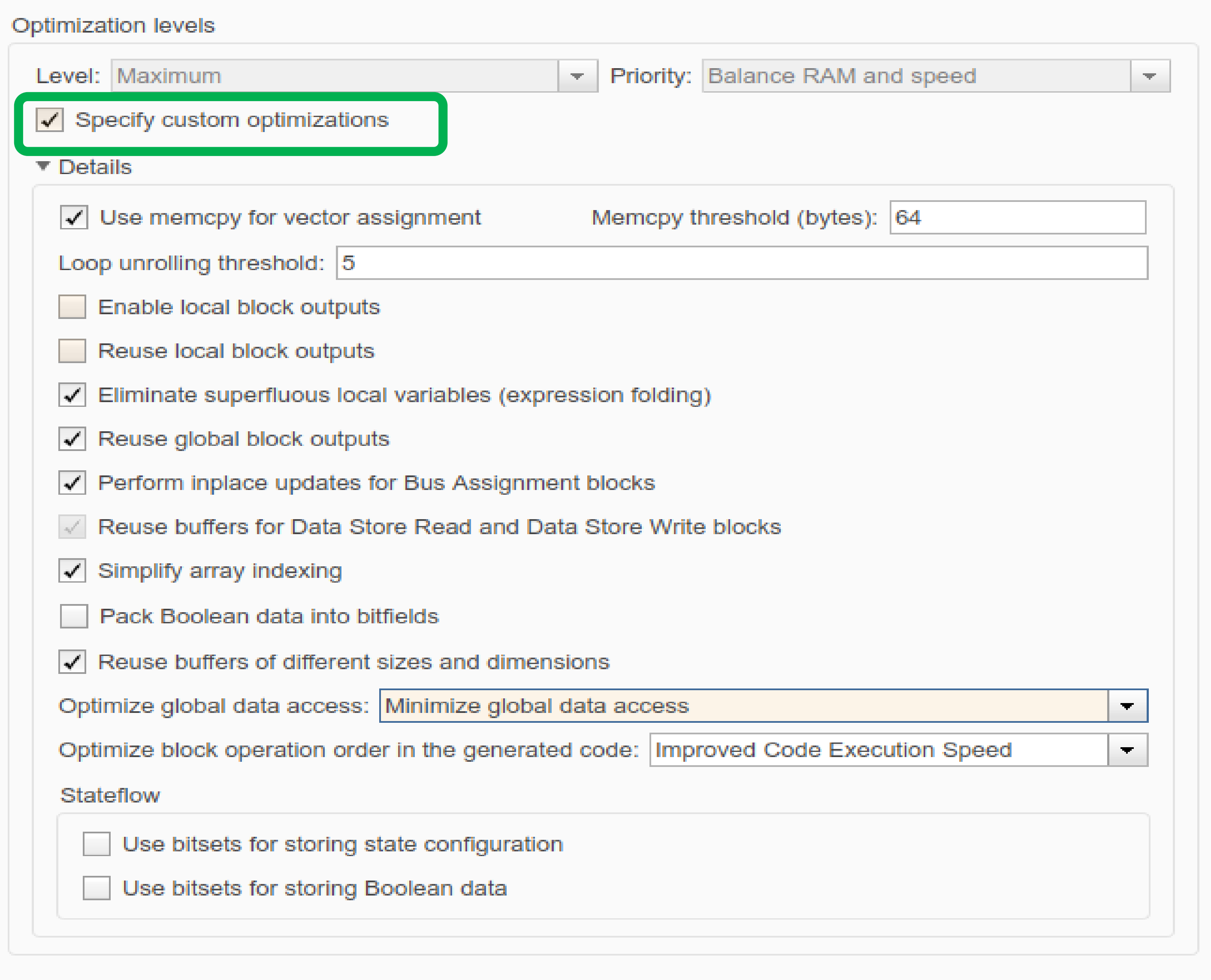
Easy Optimization Levels and Priorities



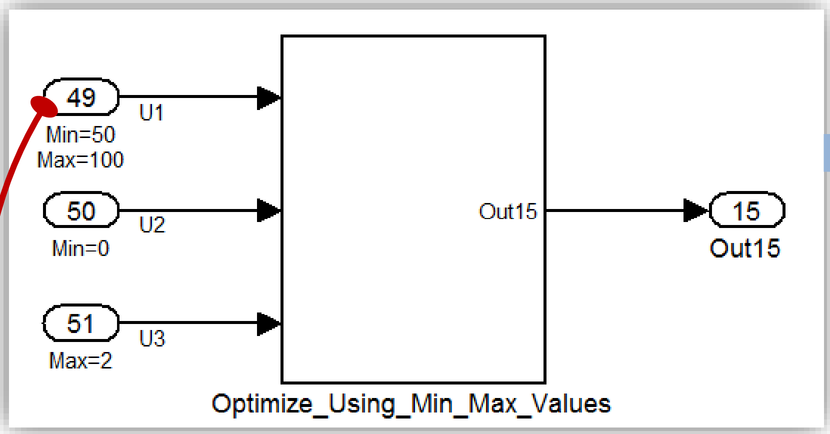
- Levels:
  - Minimum
  - Balanced with readability
  - Maximum
- Priorities
  - Balance RAM and speed
  - Maximize execution speed
  - Minimize RAM

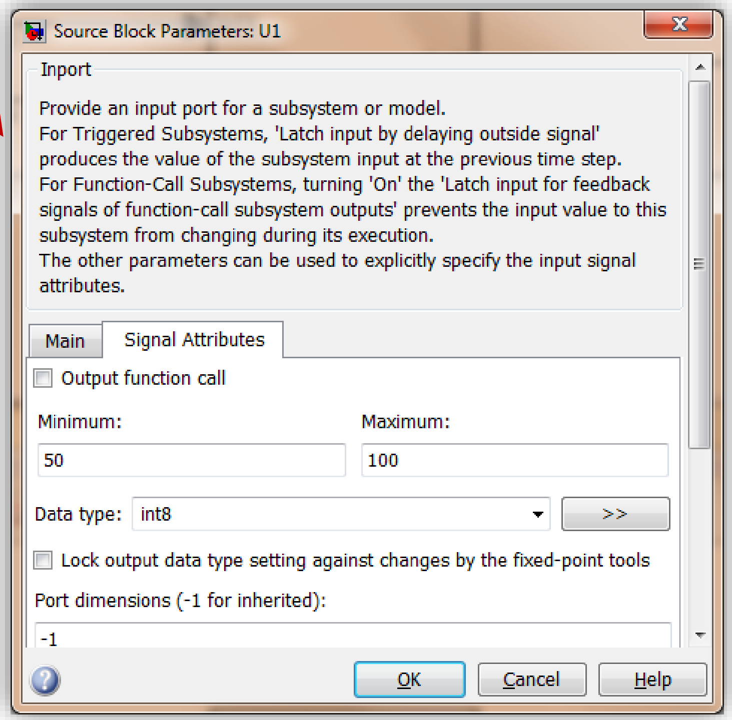
Advanced Customization

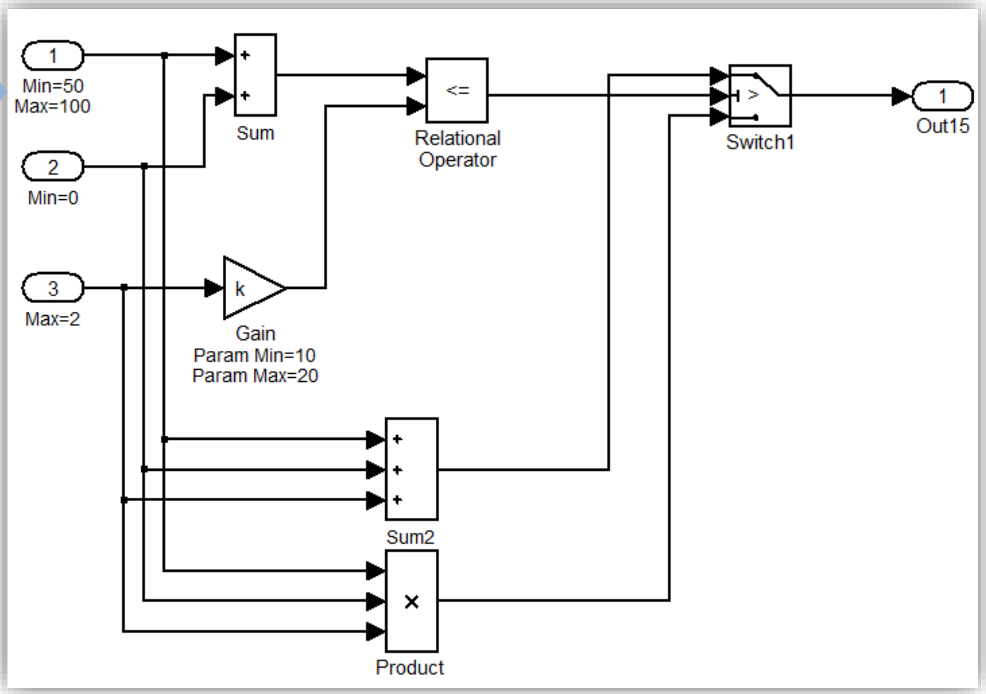
- Select individual optimizations as desired
- Preserves existing setting from previous versions



Optimize Using Min & Max Values







Optimize Using Min & Max Values

Code generation

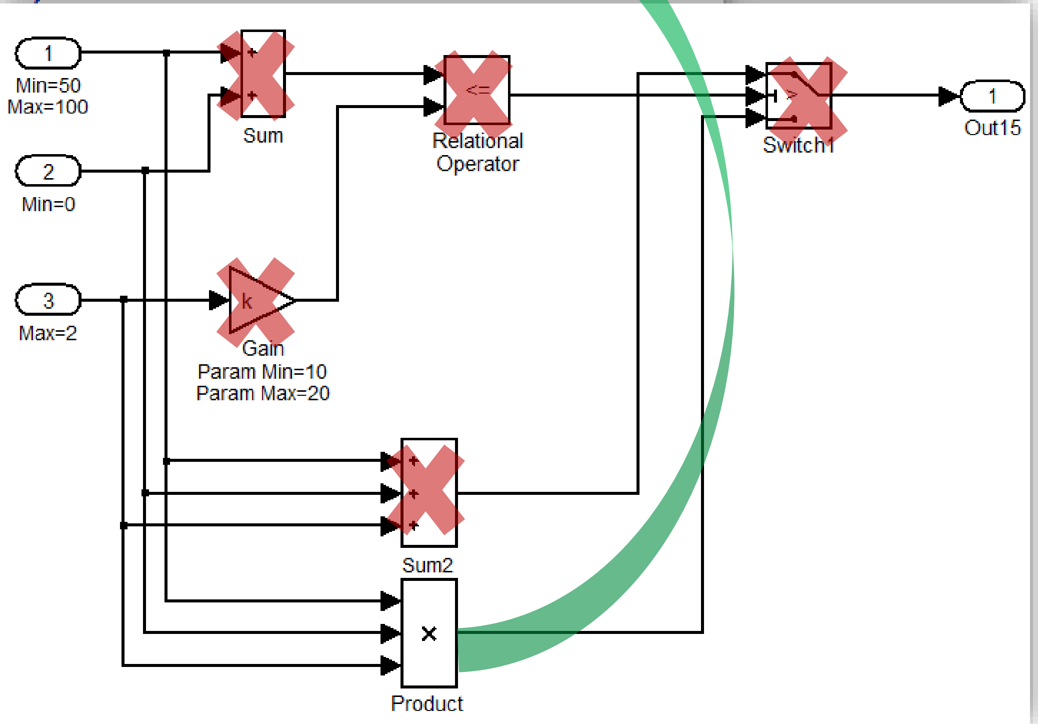
☐ Optimize using the specified minimum and maximum values

```
rtb_Sum = U1 + U2;

/* Gain: '<S8>/Gain' incorporates:
 *  Inport: '<Root>/U3'
 */
rtb_Sum2 = Code_Optimization_P.Gain_Gain * U3;

/* RelationalOperator: '<S8>/Relational Operator' */
rtb_RelationalOperator = (rtb_Sum <= rtb_Sum2);

/* Switch: '<S8>/Switch1' */
if (rtb_RelationalOperator) {
  /* Sum: '<S8>/Sum2' incorporates:
   *  Inport: '<Root>/U1'
   *  Inport: '<Root>/U2'
   *  Inport: '<Root>/U3'
   */
  rtb_Sum2 = (U1 + U2) + U3;
} else {
  /* Product: '<S8>/Product' incorporates:
   *  Inport: '<Root>/U1'
   *  Inport: '<Root>/U2'
   *  Inport: '<Root>/U3'
   */
  rtb_Sum2 = U1 * U2 * U3;
}
```

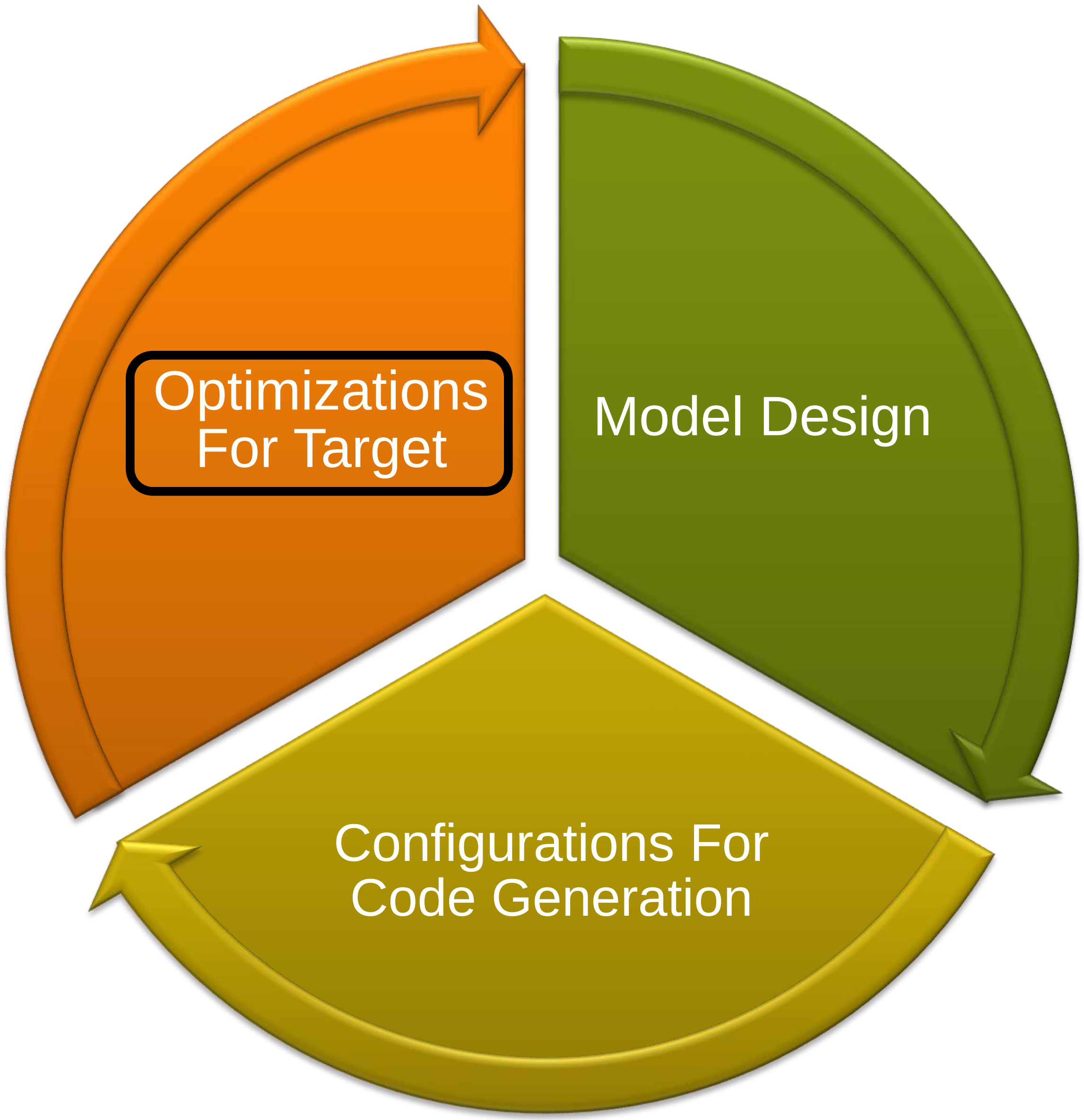
Code generation

☑ Optimize using the specified minimum and maximum values

```
/* Product: '<S8>/Product' incorporates:
 *  Inport: '<Root>/U1'
 *  Inport: '<Root>/U2'
 *  Inport: '<Root>/U3'
 *  Switch: '<S8>/Switch1'
 */
rtb_Sum2 = U1 * U2 * U3;

/* Outport: '<Root>/Out15' */
Code_Optimization_Y.Out15 = rtb_Sum2;
}
```

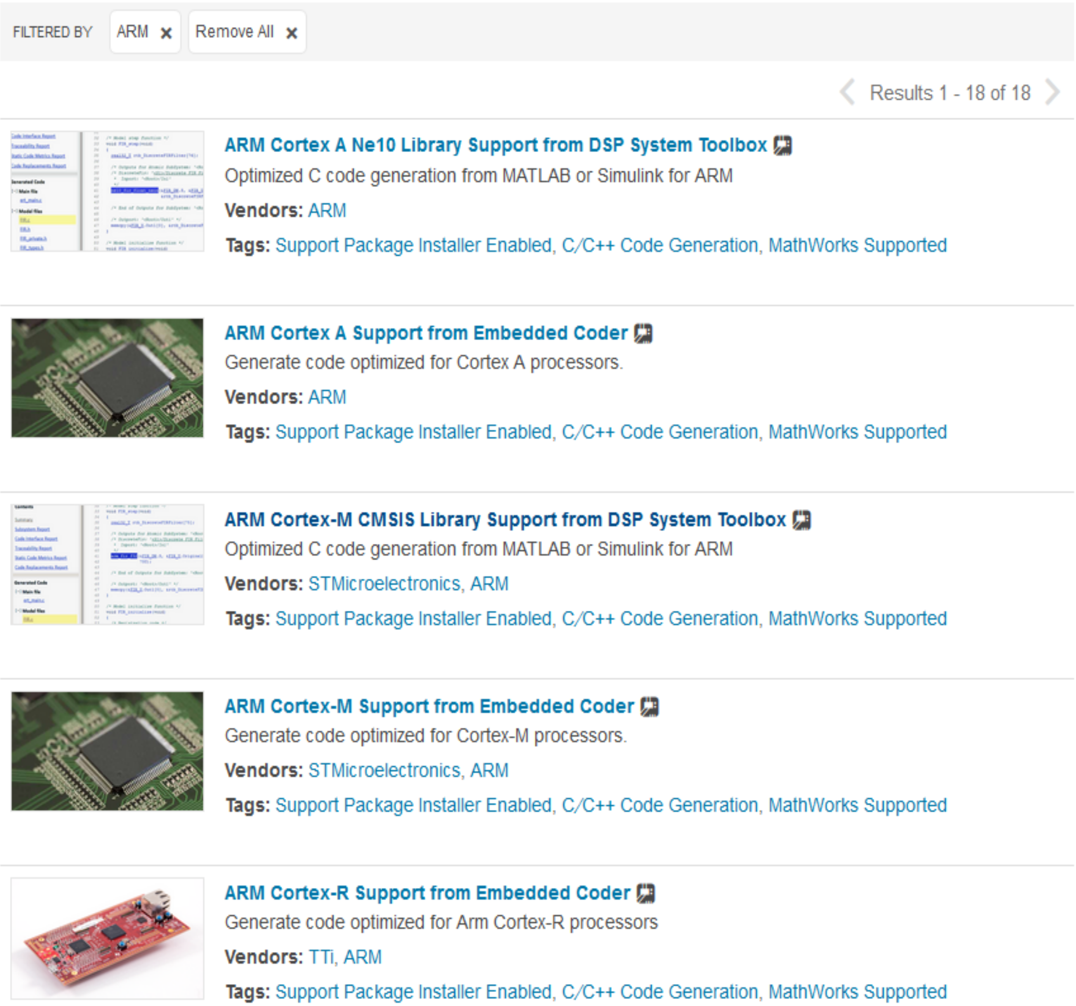


How and What Should I Optimize?

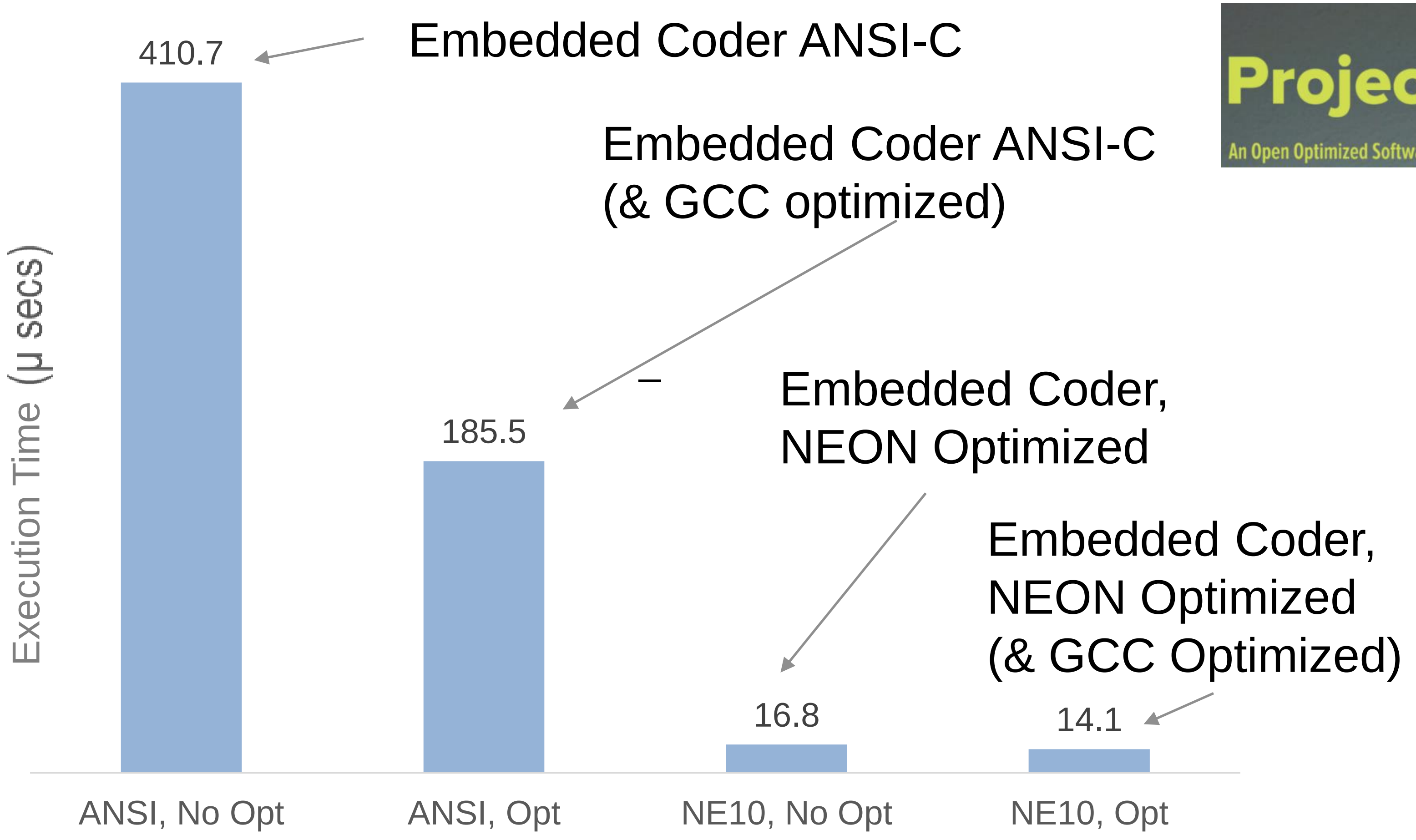


Target Based Optimizations

- Hardware Support Packages
- Code Replacement Libraries for Custom libraries eg.
  - ARM Cortex A Ne10
  - Intel SSE, AVX
  - ARM Cortex M CMSIS
- S-Functions for legacy code
- Organization wide Custom Libraries via Code Replacement Libraries



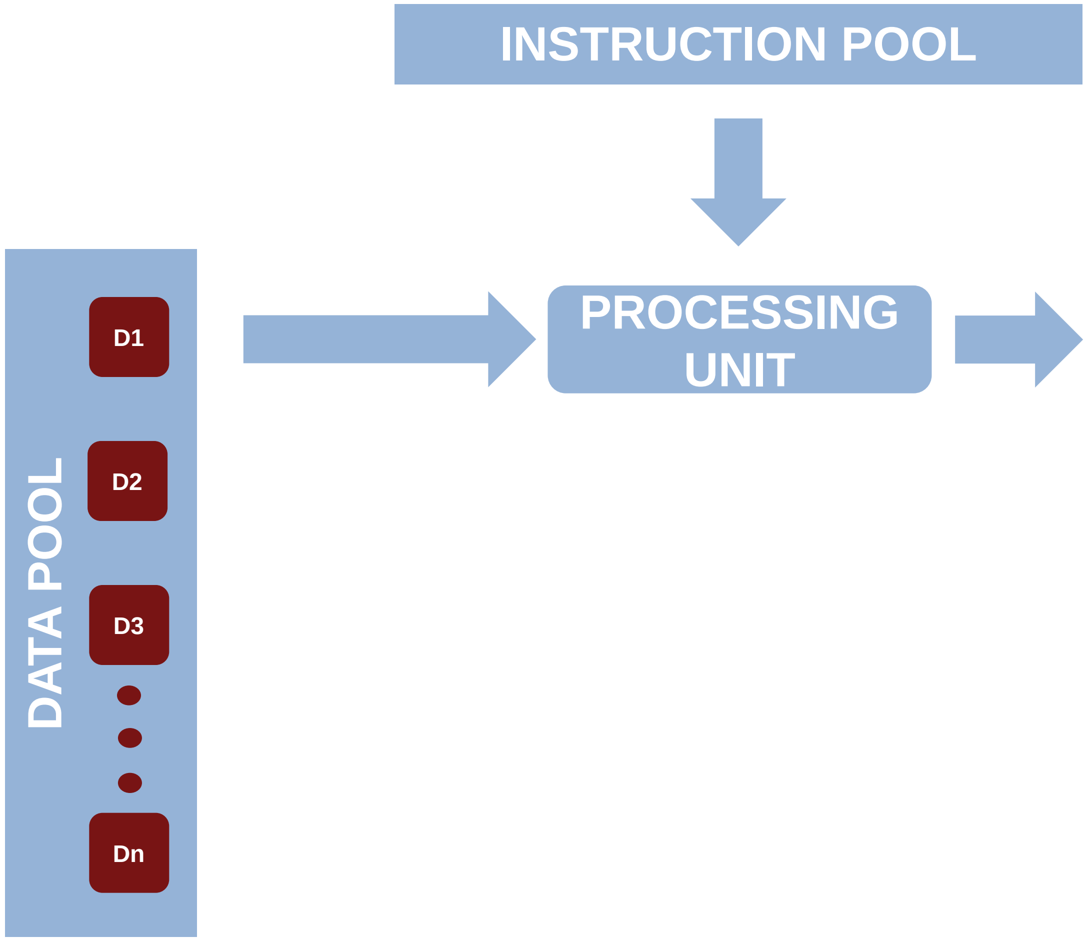
Auto Code Performance ARM Cortex-A



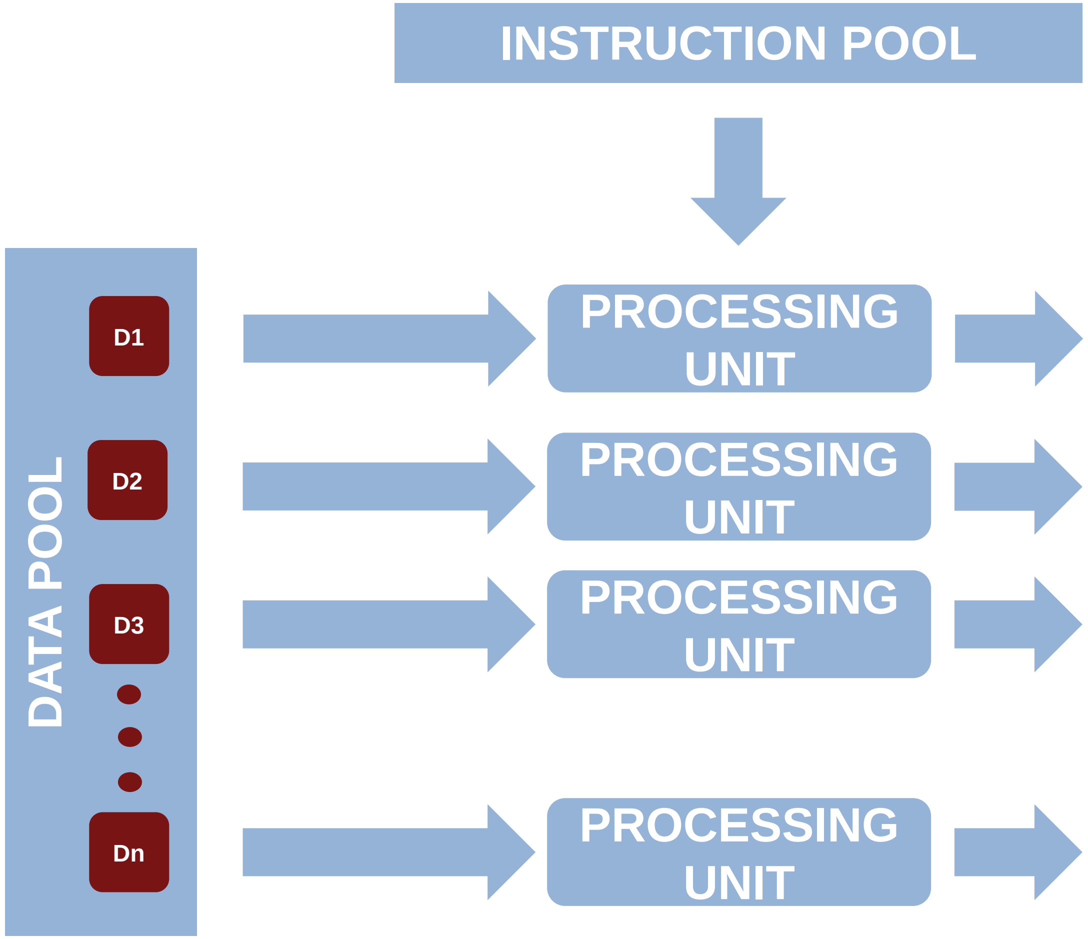
Run Format: [ANSI or Ne10], [gcc no opt or gcc -02], ARM 1Ghz Cortex A8

What is SIMD?

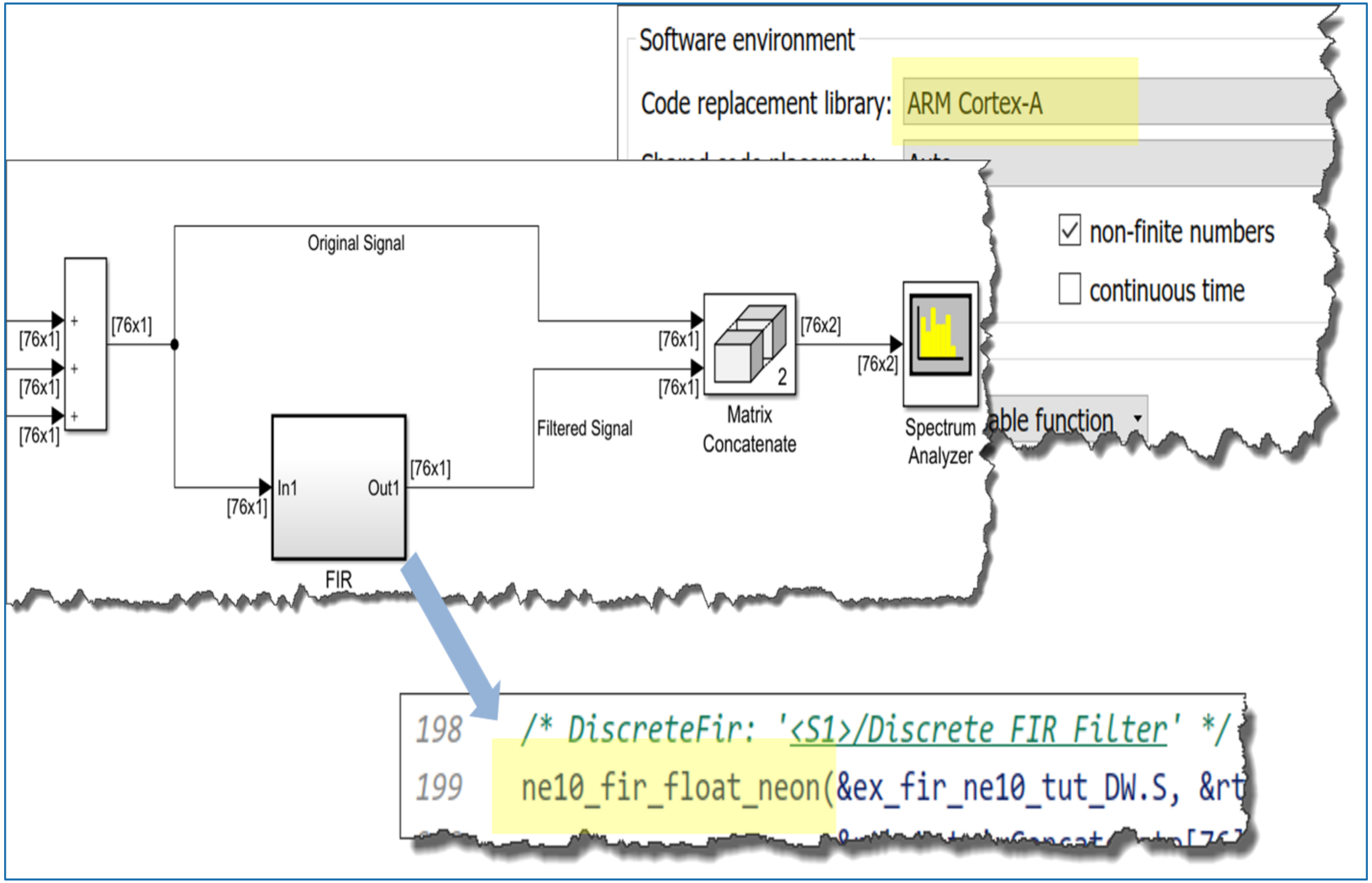
SISD - **S**ingle **I**nstruction **S**ingle **D**ata



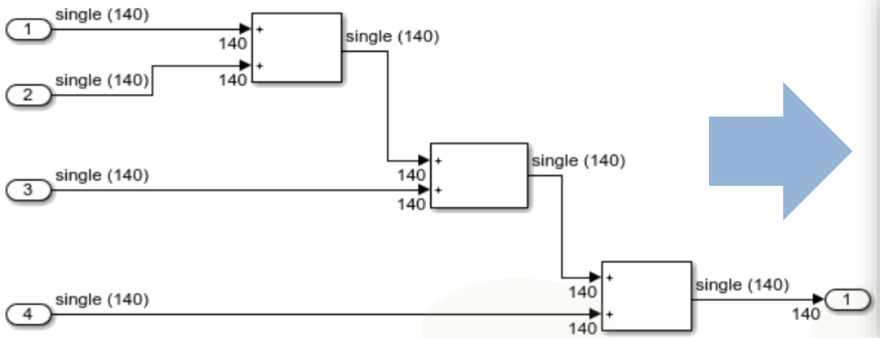
SIMD - **S**ingle **I**nstruction **M**ultiple **D**ata

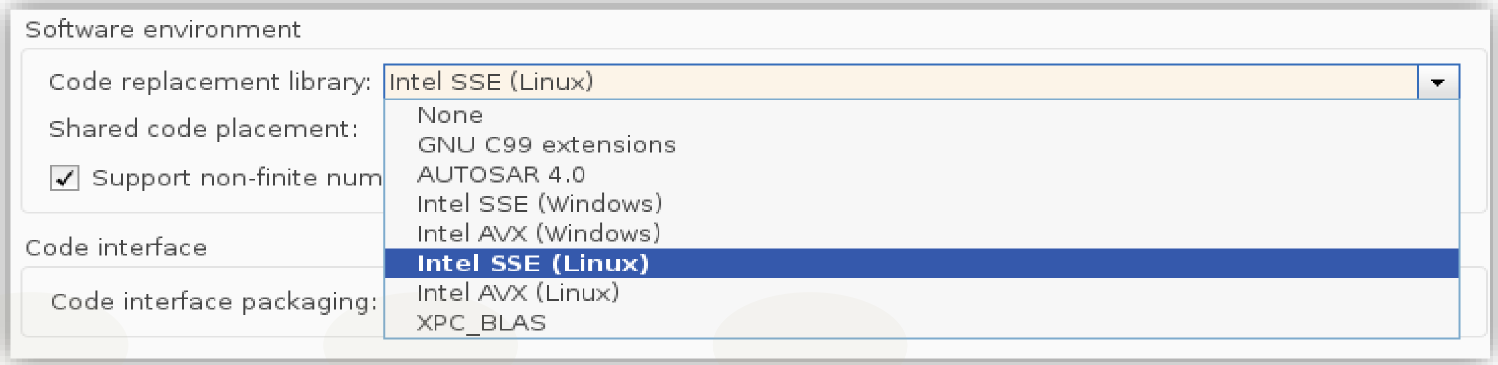


ARM Cortex-A Optimized Code



SIMD for Intel SSE/AVX using Code Replacement Library






```
void simd_model_step(void)
{
  __attribute__((aligned(16))) real32_T rtb_Add[140];
  __attribute__((aligned(16))) real32_T rtb_Add1[140];
  mw_gcc_sse_mm_add_f32x4(simd_model_U.In1, 140, 1, simd_model_U.In2, rtb_Add);
  mw_gcc_sse_mm_add_f32x4(rtb_Add, 140, 1, simd_model_U.In3, rtb_Add1);
  mw_gcc_sse_mm_add_f32x4(rtb_Add1, 140, 1, simd_model_U.In4, simd_model_Y.Out1);
}

void mw_gcc_sse_mm_add_f32x4(const float * A, int Row, int Col, const float * B, float * C)
{
  __m128 sse_a, sse_b, sse_c;
  int size = Row*Col;
  int i;
  int k=0;
  for (i = 0; i < size ; i+=4)
  {
    sse_a = _mm_load_ps(A+i);
    sse_b = _mm_load_ps(B+i);
    sse_c = _mm_add_ps(sse_a, sse_b);
    _mm_store_ps(C+i, sse_c);
  }
  k=i-4;
  for (i = 0; i < size%4 ; i++)
  {
    C[k+i] = A[k+i]+B[k+i];
  }
}
```

Wrapper function

Redundant Loops

```
void simd_model_step(void)
{
  int32_T idx;
  __m128 tmp;
  __m128 tmp_0;
  for (idx = 0; idx <= 136; idx += 4) {
    tmp = _mm_load_ps(&simd_model_U.In1[idx]);
    tmp_0 = _mm_load_ps(&simd_model_U.In2[idx]);
    tmp = _mm_add_ps(tmp, tmp_0);
    tmp_0 = _mm_load_ps(&simd_model_U.In3[idx]);
    tmp_0 = _mm_add_ps(tmp, tmp_0);
    tmp = _mm_load_ps(&simd_model_U.In4[idx]);
    tmp = _mm_add_ps(tmp_0, tmp);
    _mm_store_ps(&simd_model_Y.Out1[idx], tmp);
  }
}
```

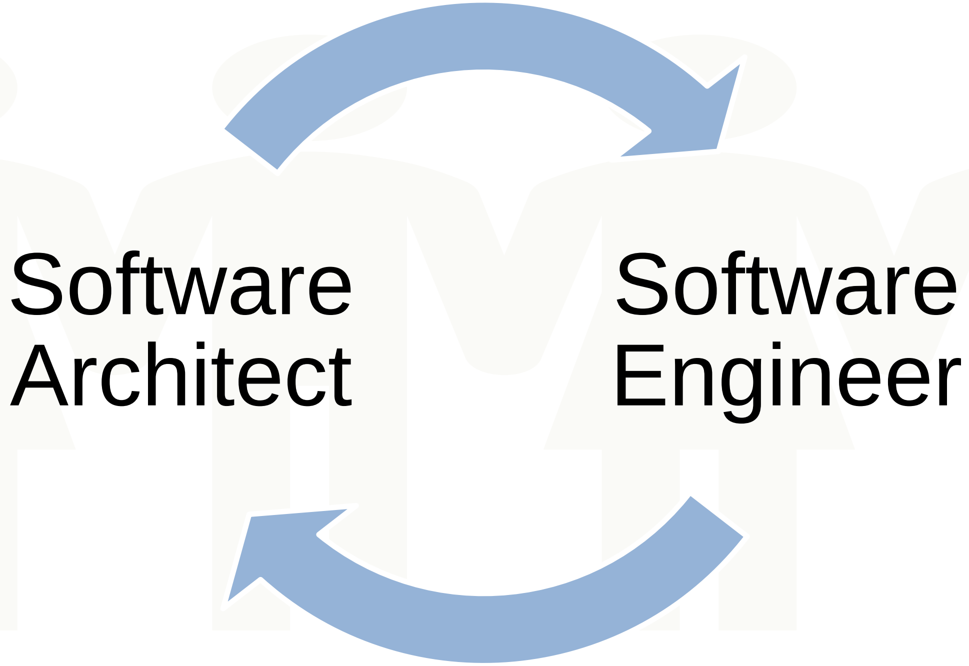
- Built on enhanced Code Replacement infrastructure
- No wrapper functions, fused loops, minimized load and stores

Join Hands & Develop

*…"*standardization *by* ***customizing*** *the Simulink® development environment"…*

Defines Architectural Details and Usage of Standard Libraries via:

- **Embedded Coder Dictionary**
- **Custom Storage Class Designer**
- **Modeling and Code Generation Best Practices**
- **Custom Replacement Libraries**
- **Target Libraries**



Configure default settings for model. Applies Software definitions, optimizations and performs verification using:

- **Code Mapping Editor**
- **Model Data Editor**
- **Function Prototype Control**
- **Model Advisor**
- **Model Optimization Configurations**
- **Simulink Data Object & Storage Classes**

Join Hands & Develop

Software Architect

Software Engineer

…Collaborative Workflow…

Quick Study

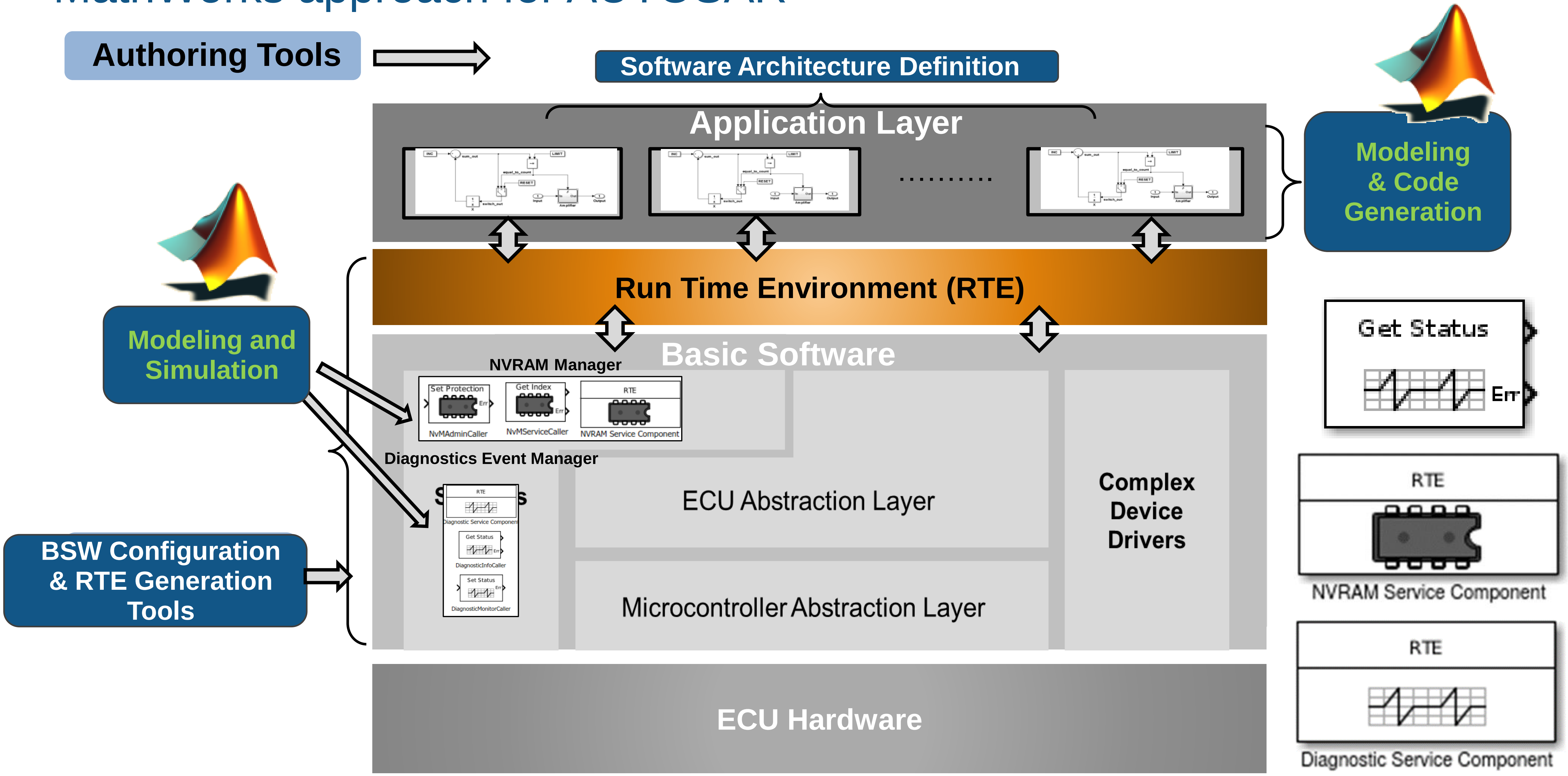
Standardize Code Architecture

Production Code Details

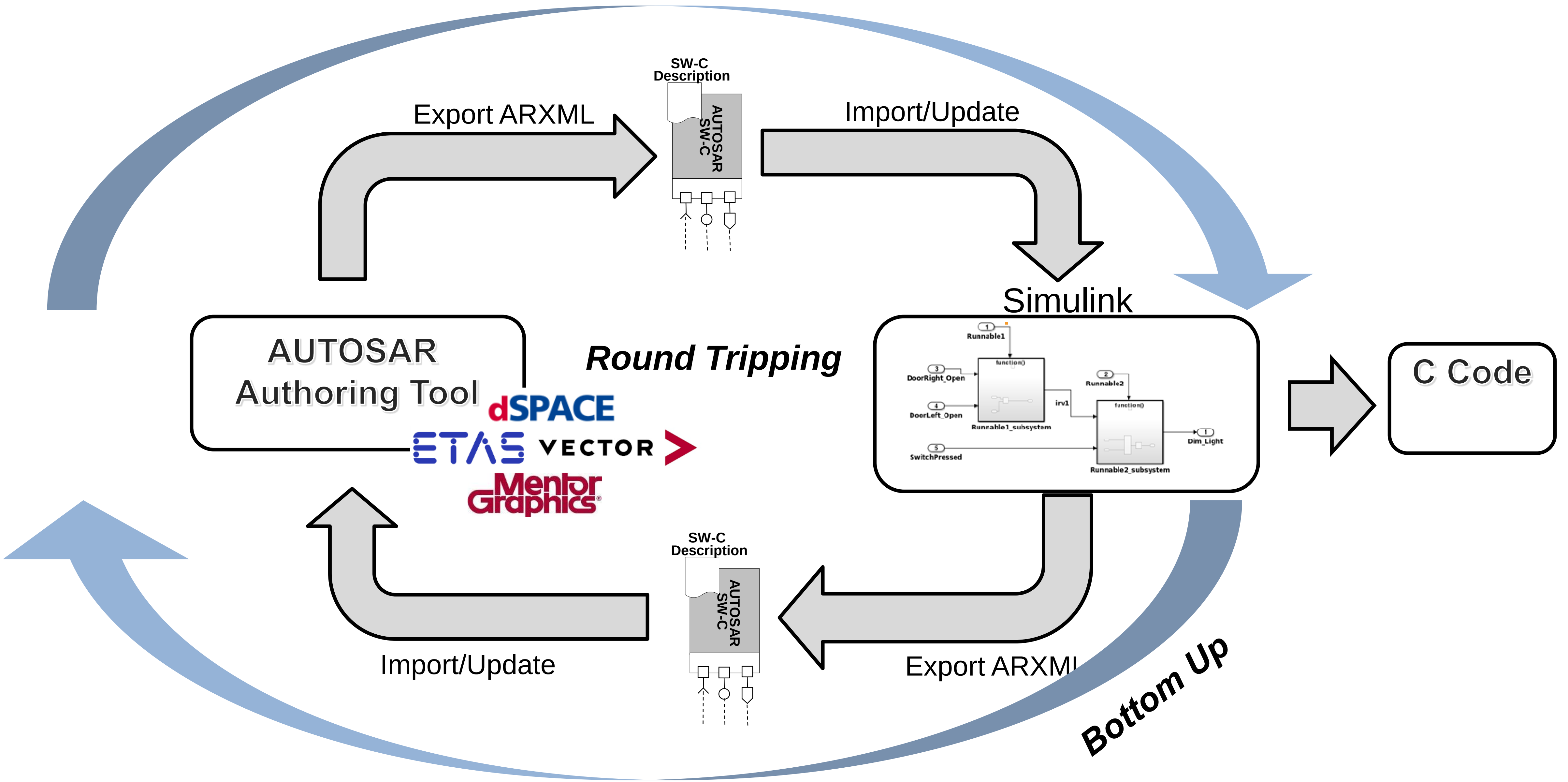
Improve Code Efficiency

Standards & Certification

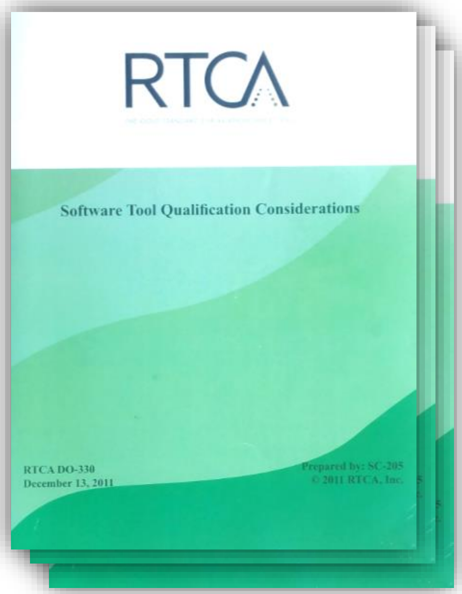
MathWorks approach for AUTOSAR



AUTOSAR Workflow



What Certification Standards Do We Support?

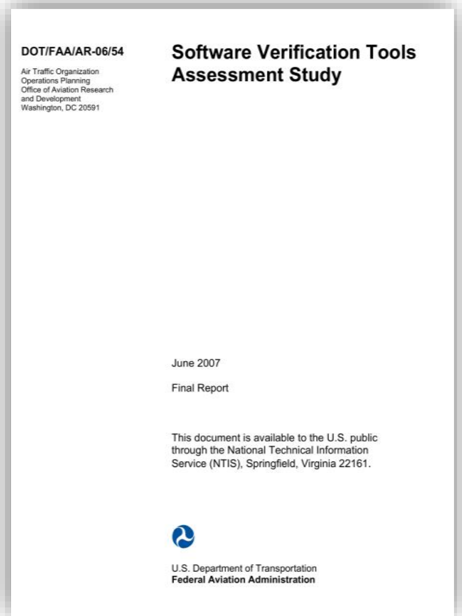


Aero:
DO-178C with Supplements and DO-254

Auto, Rail, Medical and others:
ISO 26262, EN 50128,
IEC 61508, 62304, 61511, 61131
Model Based Design & Automotive SPICE

Qual and Cert Kits

Everything Ready for Tool Qualification


DOT/FAA/AR-06/54

Air Traffic Organization
Operations Planning
Office of Aviation Research
and Development
Washington, DC 20591

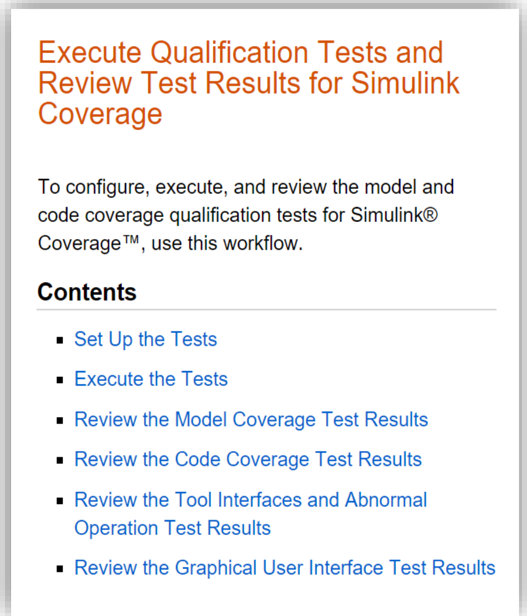
Software Verification Tools Assessment Study

June 2007

Final Report

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

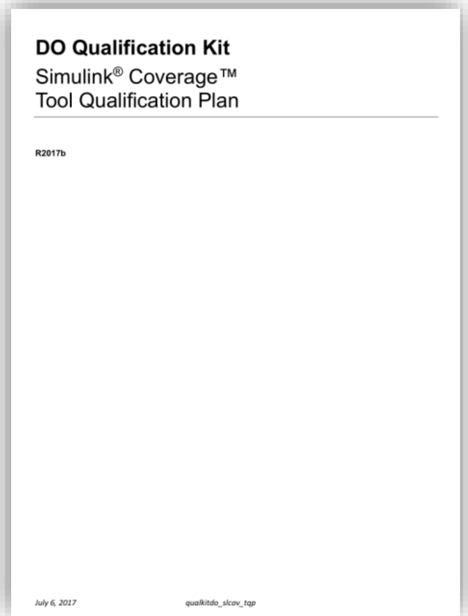
U.S. Department of Transportation
Federal Aviation Administration


Execute Qualification Tests and Review Test Results for Simulink Coverage

To configure, execute, and review the model and code coverage qualification tests for Simulink® Coverage™, use this workflow.

Contents

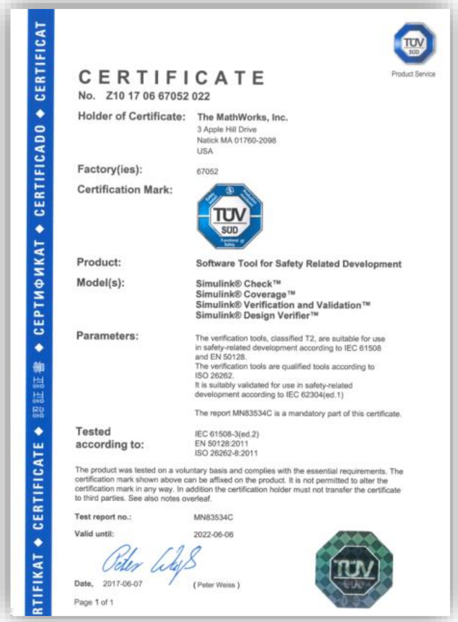
- Set Up the Tests
- Execute the Tests
- Review the Model Coverage Test Results
- Review the Code Coverage Test Results
- Review the Tool Interfaces and Abnormal Operation Test Results
- Review the Graphical User Interface Test Results


DO Qualification Kit

Simulink® Coverage™
Tool Qualification Plan

R2017b

July 6, 2017 qualkitdo_slcov_tqp


CERTIFICATE

No. Z10 17 06 67052 022

Product Service

Holder of Certificate: The MathWorks, Inc.
3 Apple Hill Drive
Natick MA 01760-2098
USA

Factory(ies): 67052

Certification Mark:

Product: Software Tool for Safety Related Development

Model(s): Simulink® Check™
Simulink® Coverage™
Simulink® Verification and Validation™
Simulink® Design Verifier™

Parameters: The verification tools, classified T2, are suitable for use in safety-related development according to IEC 61508 and EN 50128.
The verification tools are qualified tools according to ISO 26262.
It is suitably validated for use in safety-related development according to IEC 62304(ed.1)

The report MN83534C is a mandatory part of this certificate.

Tested according to: IEC 61508-3(ed.2)
EN 50128:2011
ISO 26262-8:2011

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also notes overleaf.

Test report no.: MN83534C

Valid until: 2022-06-06

Date, 2017-06-07 (Peter Weiss)

Page 1 of 1

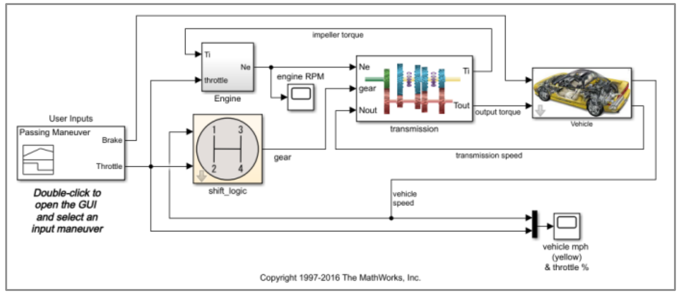
Certification Basis

Interactive Test Procedures

Artifacts and Templates

Pre-approval Certificates

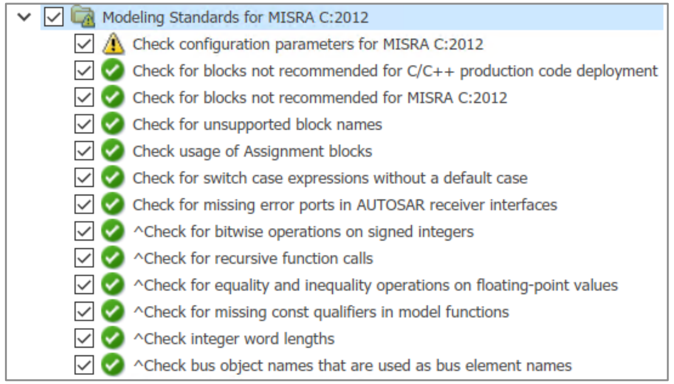
How to achieve MISRA compliance



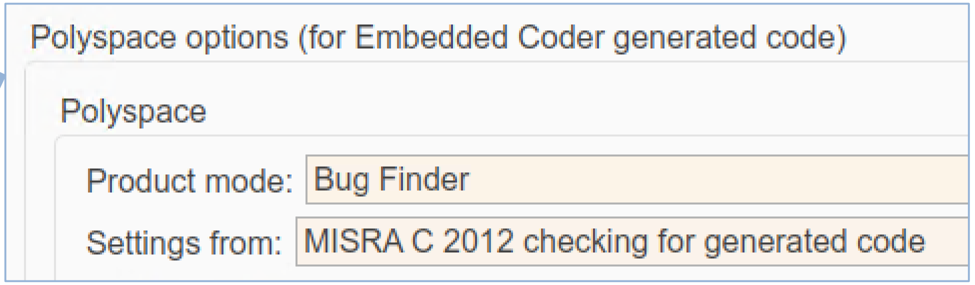
Embedded Coder

*.c / *.h

Modify or justify



Model Advisor



Polyspace Bug Finder*

*** Supports all MISRA-C 2012 Rules**

Help

MISRA C:2012 Compliance and Deviations for Code Generated using Embedded Coder

Categories
- Mandatory ☑
- Required ☑ + documented Deviations

How to achieve secure coding guideline conformance

MISRA

CERT | Software Engineering Institute | Carnegie Mellon University

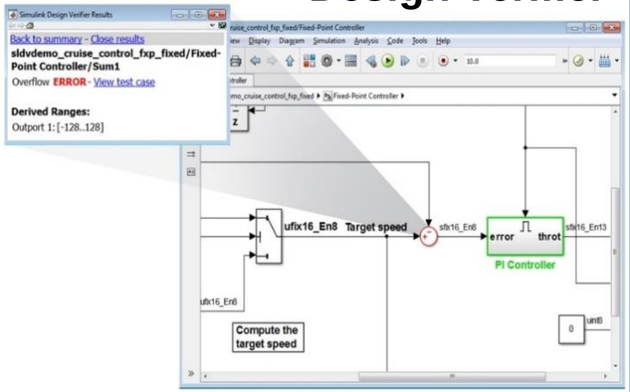
ISO/IEC TS 17961:2013
Information technology -- Programming languages, their environments and system software interfaces -- C secure coding rules

CWE **Common Weakness Enumeration**
A Community-Developed Dictionary of Software Weakness Types

Model Advisor

- Modeling Standards for MISRA C:2012
- Modeling Standards for Secure Coding (CERT C, CWE, ISO/IEC TS 17961)

Design Verifier



Polyspace Bug Finder / Code Prover

Polyspace options (for Embedded Coder generated code)

Polyspace

Product mode: Bug Finder

Settings from: MISRA C 2012 checking for generated code

Polyspace Bug Finder

Find defects: ISO-17961

Defects
- Nume
- Static
- Dynar
- Data f
- Resou

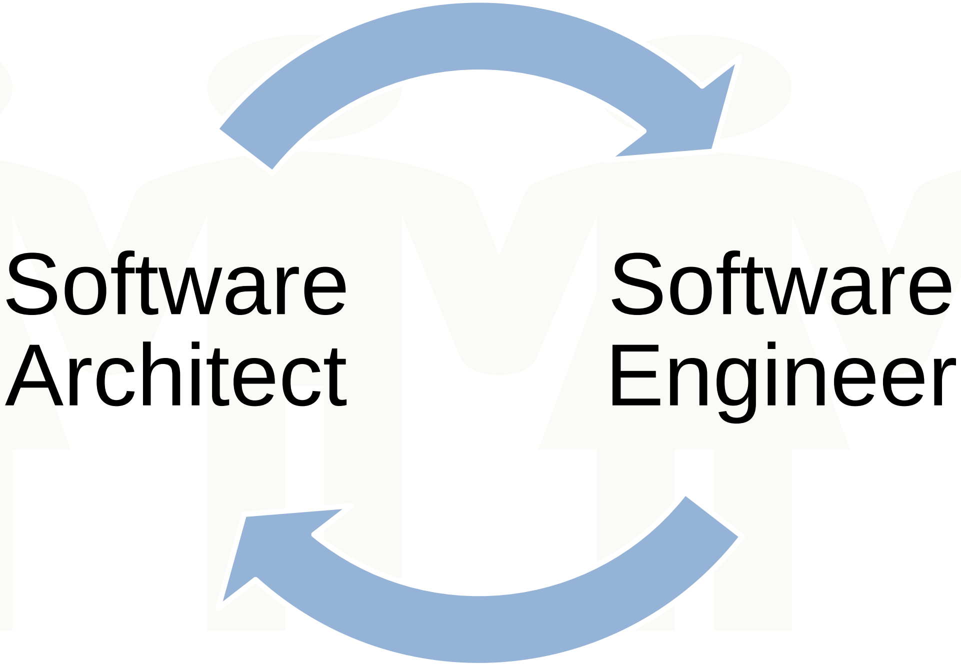
Options: default, CERT-rules, CERT-all, ISO-17961, CWE, all, custom

Join Hands & Develop

…"standardization *by **customizing** the Simulink® development environment"…*

Defines Architectural Details and Usage of Standard Libraries via:

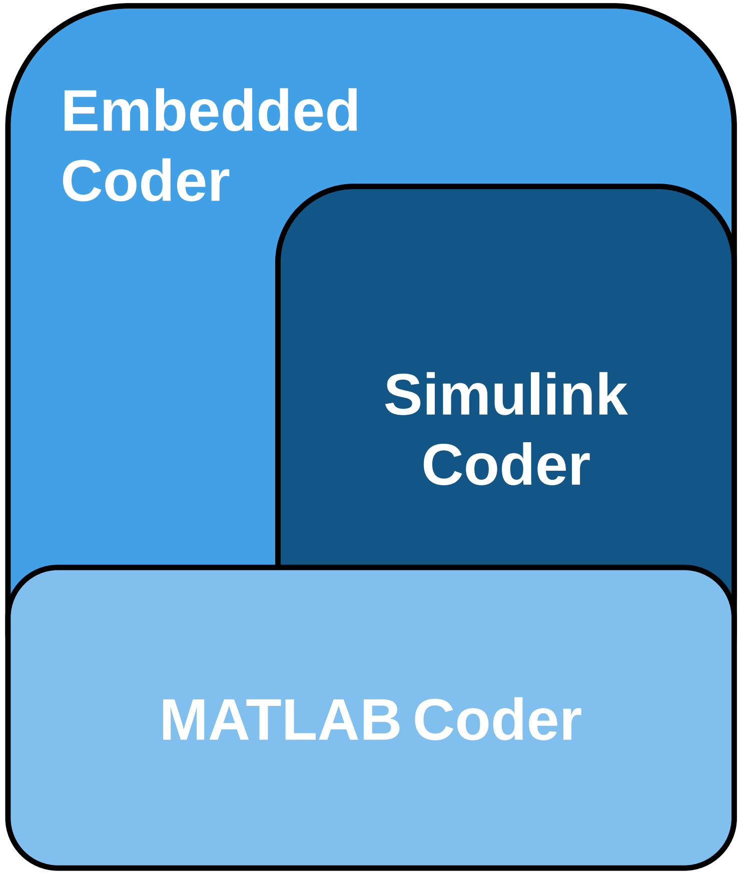
- **Embedded Coder Dictionary**
- **Custom Storage Class Designer**
- **Modeling and Code Generation Best Practices**
- **Custom Replacement Libraries**
- **Target Libraries**
- **Code Verification**



Configure default settings for model. Applies Software definitions, optimizations and performs verification using:

- **Code Mapping Editor**
- **Model Data Editor**
- **Function Prototype Control**
- **Model Advisor**
- **Model Optimization Configurations**
- **Simulink Data Object & Storage Classes**
- **Model and Code Verification**

Embedded Coder For Production Code Generation



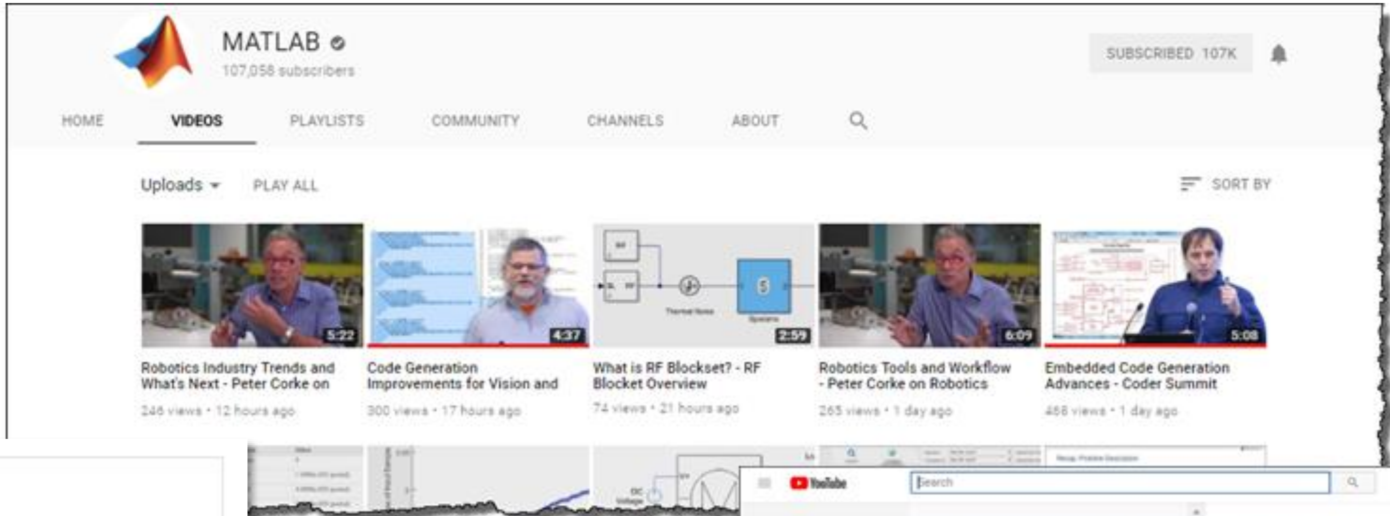
Production Code Generation with Embedded Coder

Generates C/C++ code optimized for embedded systems

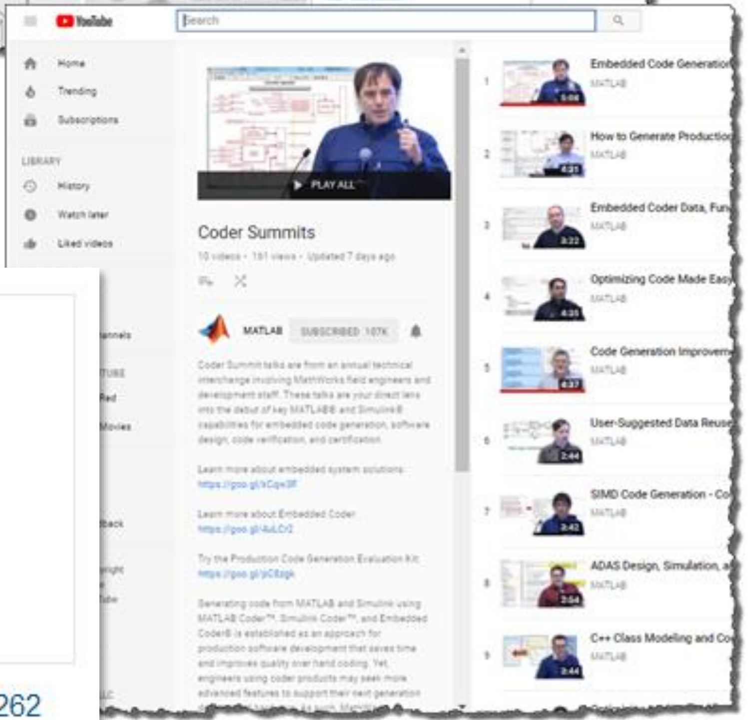
- ✓ Software-/ Processor –in-the-loop testing
- ✓ Easy integration of legacy code
- ✓ Advanced optimizations:
  - ✓ Maximize execution efficiency
  - ✓ Minimize RAM/ ROM usage
  - ✓ Target-specific function replacement
- ✓ File, Function and Interface control
- ✓ Data customization
- ✓ Support for ASAP2 and AUTOSAR
- ✓ Support for Industry Standards

Additional Resources

- Coder Summit Videos
- AUTOSAR
- ISO 26262
- DO-178
- IEC 61508
- IEC 62304
- Embedded Code Generation



BAE Systems Delivers DO-178B Level A Flight Software on Schedule with Model-Based Design



KOSTAL Asia R&D Center Receives ISO 26262 ASIL D Certification for Automotive Software Developed with Model-Based Design

LG Chem Develops AUTOSAR - and ISO 26262 - Compliant Software for a Hybrid Vehicle Battery Management System

Challenge

Design and implement production battery management system (BMS) software for the Volvo XC90 plug-in hybrid

Solution

Use Model-Based Design with MATLAB and Simulink to model, simulate, verify, and generate production code for AUTOSAR application layer software components

Results

- Existing library of core components reused
- Software issues reduced by more than 50%
- ISO 26262 ASIL C certification achieved

The LG Chem battery management system.

"Model-Based Design with MATLAB and Simulink enables us to increase component reuse, reduce manual coding, improve communication with our customers, and ultimately deliver higher-quality BMS in less time."

- Won Tae Joe, LG Chem

Link to technical article

MathWorks Training Offerings

CODE GENERATION

Fundamentals of Code Generation for Embedded Applications

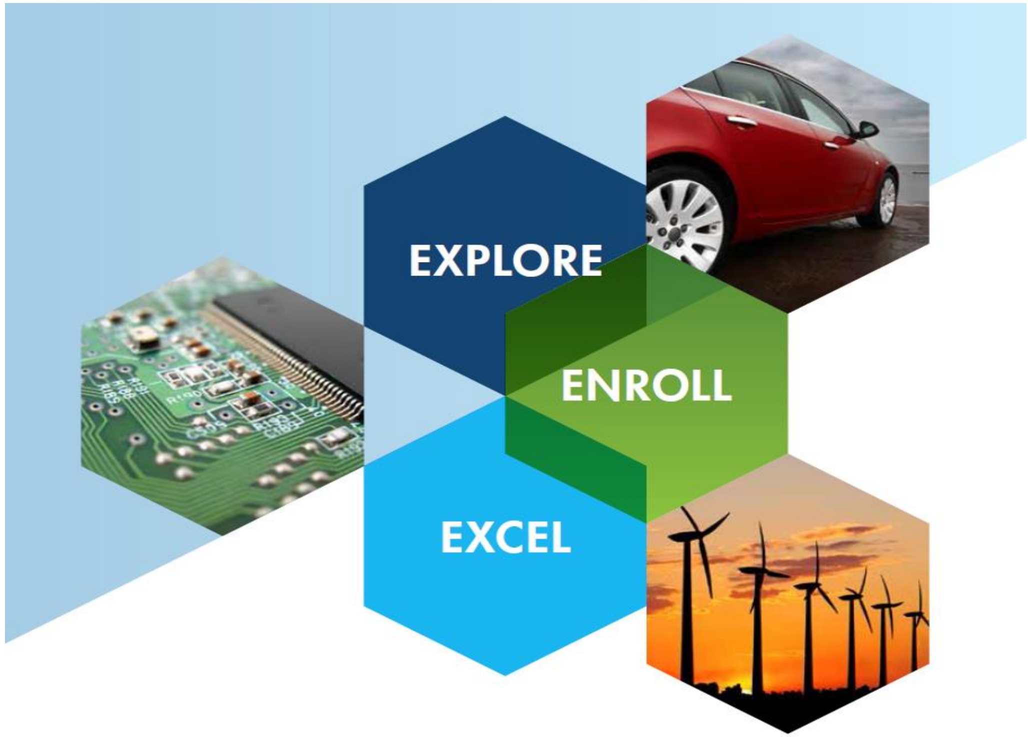
FUNDAMENTAL

- Simulation speedup with code generation
- Parameter tuning with external mode
- Code generation
- In-the-loop verification
- Code execution profiling

Embedded Coder for Production Code Generation

ADVANCED

- Generated code module and data structure
- Code generation options and optimizations
- Integrating generated code with external code
- Customizing data characteristics
- Advanced customization techniques
- Deploying embedded code



CURRICULUM PATHS

Code Generation/ Embedded Systems

Simulink for System and Algorithm Modeling → Fundamentals of Code Generation for Embedded Applications → Embedded Coder for Production Code Generation

http://www.mathworks.com/services/training/

Call To Action

Try or Buy

There are many ways to start using Embedded Coder. Download a free trial, or explore pricing and licensing options.

Get a Free Trial

Test drive Embedded Coder.

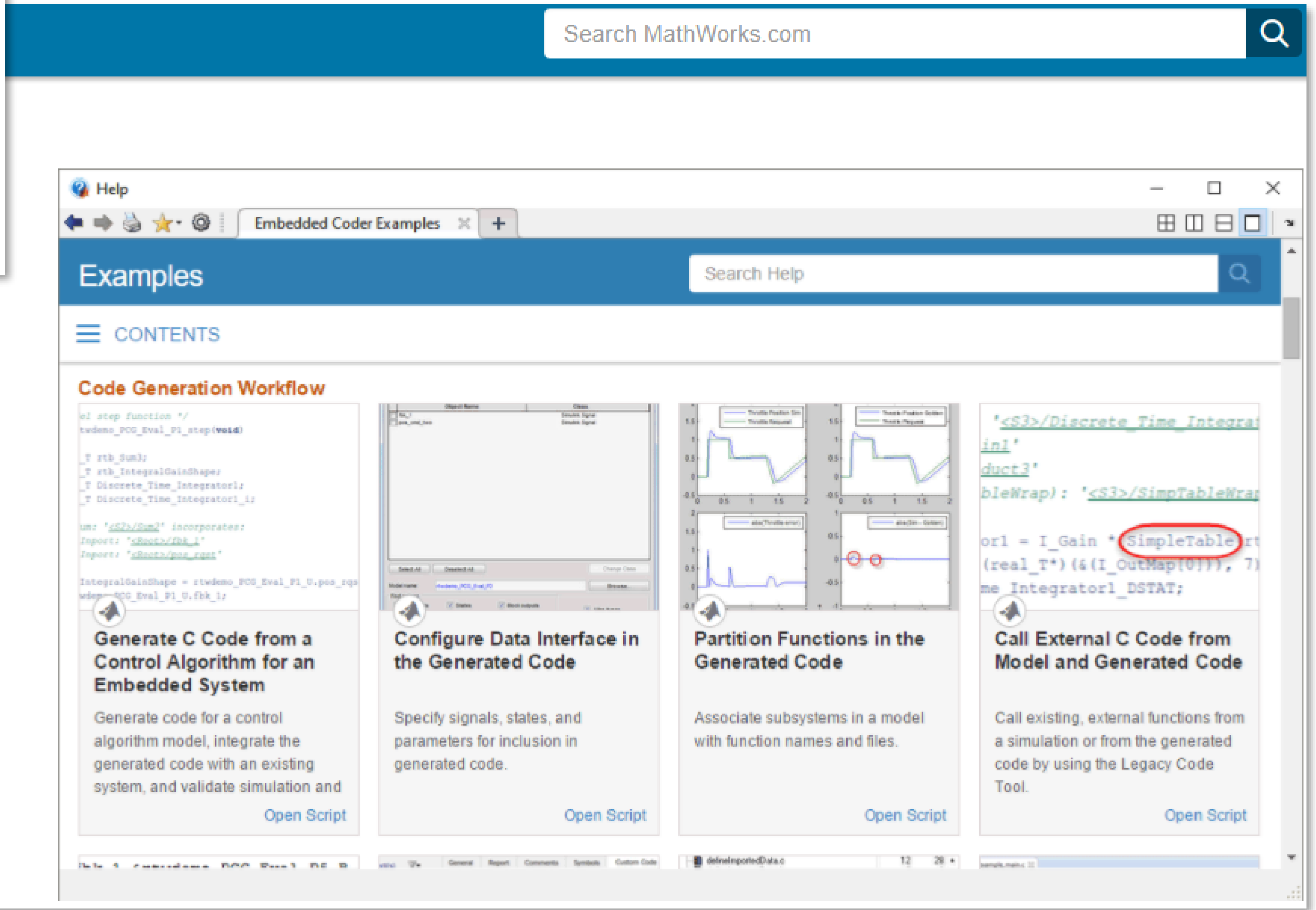
Get a trial

Ready to Buy?

Purchase Embedded Coder and explore related products.

Contact sales

Pricing and licensing



Code from Simulink

Download an interactive tutorial that guides you through the implementation of a high-level PID throttle controller to a production executable with an accompanying test harness and code metrics report. Topics include data specification, legacy code integration, and build processes.

Try the evaluation kit

[Free Production Code Generation Evaluation Kit]

Thank You ☺ !!!