Model Advisor Report – Wrong_Way_Driver_Warning.slx

Simulink version: 10.1 **Model version:** 1.133

System: Wrong_Way_Driver_Warning **Current run:** 15-Jul-2020 15:26:15

Treat as Referenced Model: off

Run Summary

Pass	Fail	Warning	Not Run	Total
✓ 344	✗ 0	⚠ 51	☰ 682	1077

By Task

Modeling Physical Systems ✓0 ✗0 ⚠0 ☰2

☰ Check consistency of block parameter units

Not Run

☰ Check for dry hydraulic nodes

Not Run

Simulink Code Inspector compatibility checks ✓0 ✗0 ⚠0 ☰69

☰ Check code generation settings

Not Run

☰ Check data import and export settings

Not Run

☰ Check diagnostic settings

Not Run

Check hardware implementation settings

Not Run

Check math and data types settings

Not Run

Check solver settings

Not Run

Check for unconnected objects in the model

Not Run

Check system target file setting

Not Run

Check function specification setting

Not Run

Check for usage of fixed-point instrumentation

Not Run

Check for unsupported blocks

Not Run

Check storage class for workspace variables

Not Run

Check GetSet storage class for workspace variables

Not Run

Check for sample times in the model

Not Run

Check usage of Sources blocks

Not Run

Check usage of Signal Routing blocks

Not Run

Check usage of Math Operations blocks

Not Run

Check usage of Signal Attributes blocks

Not Run

Check usage of Logical and Bit Operations blocks

Not Run

Check usage of Lookup Tables blocks

Not Run

Check usage of User-Defined Function blocks

Not Run

Check usage of Ports and Subsystems blocks

Not Run

Check usage of Discontinuities blocks

Not Run

Check usage of Sinks blocks

Not Run

Check usage of Discrete blocks

Not Run

Check usage of root Outport blocks

Not Run

Check for unsupported Signal Conversion blocks automatically inserted at signals entering block input ports

Not Run

Check usage of buses

Not Run

Check for usage of synthesized local data stores

Not Run

Check usage of global data stores

Not Run

Check global data stores' name shadow

Not Run

Check for root Outport blocks being conditionally assigned

Not Run

Check conditional input branch execution setting

Not Run

Check usage of Stateflow blocks

Not Run

Check for Stateflow machine data

Not Run

Check for Stateflow machine events

Not Run

Check usage of Stateflow charts

Not Run

Check usage of Stateflow data

Not Run

Check usage of Stateflow events

Not Run

Check usage of Stateflow states

Not Run

Check usage of Stateflow junctions

Not Run

Check usage of Stateflow transitions

Not Run

Check usage of Stateflow graphical functions

Not Run

Check usage of Stateflow truth tables

Not Run

Check Loop unrolling threshold setting

Not Run

Check destinations of If and Switchcase blocks

Not Run

Check for root Outport blocks that have non-auto storage class

Not Run

Check for Terminator blocks connected to Model Reference block outports

Not Run

Check for unsupported propagation of initial condition values

Not Run

Check data type replacement names

Not Run

Check usage of MATLAB Function Blocks

Not Run

Check usage of Data in MATLAB Functions

Not Run

Check usage of Code in MATLAB Functions

Not Run

Check MATLAB Code Analyzer messages

Not Run

Check for multiple sample times in model used as a model reference target

Not Run

Check Treat each discrete rate as a separate task setting

Not Run

Check model for commented out blocks

Not Run

Check model for instrumented signals

Not Run

Check model for void_void subsystems that use the same function name

Not Run

Check n-D Lookup Table blocks for incompatible breakpoint data type

Not Run

Check model for reusable subsystems that use the same function interfaces

Not Run

Check for usage of shared synthesized local data stores

Not Run

Check the code generation folder structure for the model

Not Run

Check for unsupported Code Mapping settings

Not Run

Check model for compiled and graphical block sorted order

Not Run

Check usage of String blocks

Not Run

☰ Check usage of shared utilities

Not Run

☰ Check model arguments for storage classes

Not Run

☰ Check usage of Stateflow MATLAB action language

Not Run

Modeling Standards for DO-178C/DO-331 ✓0 ✗0 ⚠0 ☰128

☰ Display model version information

Not Run

High-Integrity Systems ✓0 ✗0 ⚠0 ☰89

Simulink ✓*0* ✗*0* ⚠*0* ☰*28*

☰ Check usage of Abs blocks

Not Run

☰ Check usage of Math Function blocks (rem and reciprocal functions)

Not Run

☰ Check usage of Math Function blocks (log and log10 functions)

Not Run

☰ Check usage of While Iterator blocks

Not Run

Check usage of For and While Iterator subsystems

Not Run

Check usage of For Iterator blocks

Not Run

Check usage of If blocks and If Action Subsystem blocks

Not Run

Check usage of Switch Case blocks and Switch Case Action Subsystem blocks

Not Run

Check usage of conditionally executed subsystems

Not Run

Check usage of Merge blocks

Not Run

Check Relational Operator blocks equating floating-point types

Not Run

Check usage of Relational Operator blocks

Not Run

Check usage of Logical Operator blocks

Not Run

Check usage of bit operation blocks

Not Run

Check for blocks not recommended for C/C++ production code deployment

Not Run

Check for inconsistent vector indexing methods

Not Run

Check data types for blocks with index signals

Not Run

Check usage of variant blocks

Not Run

Check usage of lookup table blocks

Not Run

Check usage of Signal Routing blocks

Not Run

Check for root Inports with missing properties

Not Run

Check for root Inports with missing range definitions

Not Run

Check for root Outports with missing range definitions

Not Run

Check usage of Assignment blocks

Not Run

Check global variables in graphical functions

Not Run

Check usage of Gain blocks

Not Run

Check for length of user-defined object names

Not Run

Check data type of loop control variables

Not Run

Stateflow ✓*0* ✗*0* ⚠*0* ☰*14*

Check state machine type of Stateflow charts

Not Run

Check Stateflow charts for ordering of states and transitions

Not Run

Check usage of bitwise operations in Stateflow charts

Not Run

Check for Strong Data Typing with Simulink I/O

Not Run

Check Stateflow debugging options

Not Run

Check Stateflow charts for transition paths that cross parallel state boundaries

Not Run

Check for inappropriate use of transition paths

Not Run

Check Stateflow charts for strong data typing

Not Run

Check naming of ports in Stateflow charts

Not Run

Check scoping of Stateflow data objects

Not Run

Check Stateflow charts for uniquely defined data objects

Not Run

Check usage of shift operations for Stateflow data

Not Run

Check assignment operations in Stateflow charts

Not Run

Check Stateflow charts for unary operators

Not Run

MATLAB ✓*0* ✗*0* ⚠*0* ☰*10*

Check usage of standardized MATLAB function headers

Not Run

Check for MATLAB Function interfaces with inherited properties

Not Run

Check MATLAB Function metrics

Not Run

Check MATLAB Code Analyzer messages

Not Run

Check if/elseif/else patterns in MATLAB Function blocks

Not Run

Check switch statements in MATLAB Function blocks

Not Run

Check usage of relational operators in MATLAB Function blocks

Not Run

Check usage of equality operators in MATLAB Function blocks

Not Run

Check usage of logical operators and functions in MATLAB Function blocks

Not Run

Check type and size of condition expressions

Not Run

Configuration ✓*0* ✗*0* ⚠*0* ☰*32*

Check safety-related diagnostic settings for data store memory

Not Run

Check safety-related diagnostic settings for saving

Not Run

Check safety-related model referencing settings

Not Run

Check safety-related code generation settings for comments

Not Run

Check safety-related code generation interface settings

Not Run

Check safety-related solver settings for simulation time

Not Run

Check safety-related solver settings for solver options

Not Run

Check safety-related solver settings for tasking and sample-time

Not Run

Check safety-related diagnostic settings for solvers

Not Run

Check safety-related diagnostic settings for sample time

Not Run

Check safety-related optimization settings for logic signals

Not Run

Check safety-related block reduction optimization settings

Not Run

Check safety-related code generation settings for code style

Not Run

Check safety-related optimization settings for application lifespan

Not Run

Check safety-related code generation identifier settings

Not Run

Check safety-related optimization settings for loop unrolling threshold

Not Run

Check safety-related optimization settings for data initialization

Not Run

Check safety-related optimization settings for data type conversions

Not Run

Check safety-related optimization settings for division arithmetic exceptions

Not Run

Check safety-related optimization settings for specified minimum and maximum values

Not Run

Check safety-related diagnostic settings for compatibility

Not Run

Check safety-related diagnostic settings for parameters

Not Run

Check safety-related diagnostic settings for Merge blocks

Not Run

Check safety-related diagnostic settings for model initialization

Not Run

Check safety-related diagnostic settings for data used for debugging

Not Run

Check safety-related diagnostic settings for signal connectivity

Not Run

Check safety-related diagnostic settings for bus connectivity

Not Run

Check safety-related diagnostic settings that apply to function-call connectivity

Not Run

Check safety-related diagnostic settings for type conversions

Not Run

Check safety-related diagnostic settings for model referencing

Not Run

Check safety-related diagnostic settings for Stateflow

Not Run

Check safety-related diagnostic settings for signal data

Not Run

Naming ✓*0* ✗*0* ⚠*0* ☰*2*

Check model file name

Not Run

Check model object names

Not Run

Requirements ✓*0* ✗*0* ⚠*0* ☰*1*

☰ Check for model elements that do not link to requirements

Not Run

Code ✓*0* ✗*0* ⚠*0* ☰*2*

☰ Check for blocks not recommended for MISRA C:2012

Not Run

☰ Check configuration parameters for MISRA C:2012

Not Run

Simulink ✓0 ✗0 ⚠0 ☰9

☰ *Check for Discrete-Time Integrator blocks with initial condition uncertainty*

Not Run

☰ *Check root model Inport block specifications*

Not Run

☰ *Identify unconnected lines, input ports, and output ports*

Not Run

☰ *Check usage of tunable parameters in blocks*

Not Run

☰ *Check for blocks that have constraints on tunable parameters*

Not Run

☰ *Identify questionable subsystem settings*

Not Run

Check bus signals treated as vectors

Not Run

Check for potentially delayed function-call block return values

Not Run

Check usage of Merge blocks

Not Run

Stateflow ✓0 ✗0 ⚠0 ☰3

Check for Strong Data Typing with Simulink I/O

Not Run

Check definition of Stateflow data

Not Run

Check usage of exclusive and default states in state machines

Not Run

Library Links ✓0 ✗0 ⚠0 ☰3

Identify disabled library links

Not Run

Identify parameterized library links

Not Run

Identify unresolved library links

Not Run

Model Referencing ✓0 ✗0 ⚠0 ▤2

▤ *Check for model reference configuration mismatch*

Not Run

▤ *Check for parameter tunability information ignored for referenced models*

Not Run

Requirements Consistency ✓0 ✗0 ⚠0 ▤4

▤ *Identify requirement links that specify invalid locations within documents*

Not Run

▤ *Identify requirement links with missing documents*

Not Run

▤ *Identify requirement links with path type inconsistent with preferences*

Not Run

▤ *Identify selection-based links having description fields that do not match their requirements document text*

Not Run

Simulink Coder ✓0 ✗0 ⚠0 ▤3

▤ *Check sample times and tasking mode*

Not Run

▤ *Check solver for code generation*

Not Run

Check the hardware implementation

Not Run

Bug Reports ✓0 ✗0 ⚠0 ▤14

Display bug reports for DO Qualification Kit

Not Run

Display bug reports for Embedded Coder

Not Run

Display bug reports for Polyspace Code Prover

Not Run

Display bug reports for Polyspace Code Prover Server

Not Run

Display bug reports for Polyspace Bug Finder

Not Run

Display bug reports for Polyspace Bug Finder Server

Not Run

Display bug reports for Simulink Code Inspector

Not Run

Display bug reports for Simulink Report Generator

Not Run

Display bug reports for Simulink Check

Not Run

Display bug reports for Simulink Coverage

Not Run

Display bug reports for Simulink Test

Not Run

Display bug reports for Simulink Design Verifier

Not Run

Display bug reports for Simulink Requirements

Not Run

Display bug reports for Simulink

Not Run

Modeling Standards for DO-254 ✓0 ✗0 ⚠0 ☰91

Display model version information

Not Run

High-Integrity Systems ✓0 ✗0 ⚠0 ☰48

Simulink ✓*0* ✗*0* ⚠*0* ☰*16*

Check usage of Abs blocks

Not Run

Check usage of conditionally executed subsystems

Not Run

Check Relational Operator blocks equating floating-point types

Not Run

Check usage of Relational Operator blocks

Not Run

Check usage of Logical Operator blocks

Not Run

Check usage of bit operation blocks

Not Run

Check for inconsistent vector indexing methods

Not Run

Check data types for blocks with index signals

Not Run

Check for root Inports with missing properties

Not Run

Check for root Inports with missing range definitions

Not Run

Check for root Outports with missing range definitions

Not Run

Check usage of Assignment blocks

Not Run

Check global variables in graphical functions

Not Run

Check usage of Gain blocks

Not Run

Check for length of user-defined object names

Not Run

Check data type of loop control variables

Not Run

Stateflow ✓*0* ✗*0* ⚠*0* ▤*11*

Check Stateflow charts for ordering of states and transitions

Not Run

Check usage of bitwise operations in Stateflow charts

Not Run

Check for Strong Data Typing with Simulink I/O

Not Run

Check Stateflow debugging options

Not Run

Check Stateflow charts for transition paths that cross parallel state boundaries

Not Run

Check for inappropriate use of transition paths

Not Run

Check naming of ports in Stateflow charts

Not Run

Check scoping of Stateflow data objects

Not Run

Check Stateflow charts for uniquely defined data objects

Not Run

Check usage of shift operations for Stateflow data

Not Run

Check Stateflow charts for unary operators

Not Run

MATLAB ✓*0* ✗*0* ⚠*0* ☰*7*

Check usage of standardized MATLAB function headers

Not Run

Check MATLAB Code Analyzer messages

Not Run

Check if/elseif/else patterns in MATLAB Function blocks

Not Run

Check switch statements in MATLAB Function blocks

Not Run

Check usage of relational operators in MATLAB Function blocks

Not Run

Check usage of equality operators in MATLAB Function blocks

Not Run

Check usage of logical operators and functions in MATLAB Function blocks

Not Run

Configuration ✓*0* ✗*0* ⚠*0* ▤*11*

Check safety-related diagnostic settings for saving

Not Run

Check safety-related model referencing settings

Not Run

Check safety-related block reduction optimization settings

Not Run

Check safety-related diagnostic settings for parameters

Not Run

Check safety-related diagnostic settings for model initialization

Not Run

Check safety-related diagnostic settings for signal connectivity

Not Run

Check safety-related diagnostic settings for bus connectivity

Not Run

Check safety-related diagnostic settings for type conversions

Not Run

Check safety-related diagnostic settings for model referencing

Not Run

Check safety-related diagnostic settings for Stateflow

Not Run

Check safety-related diagnostic settings for signal data

Not Run

Naming ✓*0* ✗*0* ⚠*0* ▤*2*

Check model file name

Not Run

Check model object names

Not Run

Requirements ✓*0* ✗*0* ⚠*0* ▤*1*

Check for model elements that do not link to requirements

Not Run

Library Links ✓0 ✗0 ⚠0 ▤3

Identify disabled library links

Not Run

Identify parameterized library links

Not Run

Identify unresolved library links

Not Run

Model Referencing ✓0 ✗0 ⚠0 ▤1

▤ *Check for model reference configuration mismatch*

Not Run

📁 Requirements Consistency ✓0 ✗0 ⚠0 ▤4

▤ *Identify requirement links that specify invalid locations within documents*

Not Run

▤ *Identify requirement links with missing documents*

Not Run

▤ *Identify requirement links with path type inconsistent with preferences*

Not Run

▤ *Identify selection-based links having description fields that do not match their requirements document text*

Not Run

📁 HDL Coder ✓0 ✗0 ⚠0 ▤34

📁 *Checks for blocks and block settings* ✓*0* ✗*0* ⚠*0* ▤*7*

▤ Check for infinite and continuous sample time sources

Not Run

▤ Check for unsupported blocks

Not Run

▤ Check for MATLAB Function block settings

Not Run

Check for Stateflow chart settings

Not Run

Check for obsolete Unit Delay Enabled/Resettable blocks

Not Run

Check for unsupported storage class for signal objects

Not Run

Check for large matrix operations

Not Run

Industry standard checks ✓*0* ✗*0* ⚠*0* ▤*11*

Check VHDL file extension

Not Run

Check naming conventions

Not Run

Check top-level subsystem/port names

Not Run

Check module/entity names

Not Run

Check package file names

Not Run

Check signal and port names

Not Run

Check generics

Not Run

Check clock, reset, and enable signals

Not Run

Check architecture name

Not Run

Check entity and architecture

Not Run

Check clock settings

Not Run

Model configuration checks ✓*0* ✗*0* ⚠*0* ☰*6*

Check for safe model parameters

Not Run

Check for global reset setting for Xilinx and Altera devices

Not Run

Check inline configurations setting

Not Run

Check for visualization settings

Not Run

Check delay balancing setting

Not Run

Check algebraic loops

Not Run

Native Floating Point checks ✓*0* ✗*0* ⚠*0* ▤*8*

Check for blocks with nonzero output latency

Not Run

Check blocks with nonzero ulp error

Not Run

Check for single datatypes in the model

Not Run

Check for double datatypes in the model with Native Floating Point

Not Run

Check for Data Type Conversion blocks with incompatible settings

Not Run

Check for HDL Reciprocal block usage

Not Run

Check for Relational Operator block usage

Not Run

Check for unsupported blocks with Native Floating Point

Not Run

Checks for ports and subsystems ✓*0* ✗*0* ⚠*0* ▤*2*

Check for invalid top level subsystem

Not Run

Check initial conditions of enabled and triggered subsystems

Not Run

Modeling Standards for IEC 61508 ✓0 ✗0 ⚠0 ☰106

Display configuration management data

Not Run

Display model metrics and complexity report

Not Run

Check for unconnected objects

Not Run

High-Integrity Systems ✓0 ✗0 ⚠0 ☰89

Simulink ✓0 ✗0 ⚠0 ☰28

Check usage of Abs blocks

Not Run

Check usage of Math Function blocks (rem and reciprocal functions)

Not Run

Check usage of Math Function blocks (log and log10 functions)

Not Run

Check usage of While Iterator blocks

Not Run

Check usage of For and While Iterator subsystems

Not Run

Check usage of For Iterator blocks

Not Run

Check usage of If blocks and If Action Subsystem blocks

Not Run

Check usage of Switch Case blocks and Switch Case Action Subsystem blocks

Not Run

Check usage of conditionally executed subsystems

Not Run

Check usage of Merge blocks

Not Run

Check Relational Operator blocks equating floating-point types

Not Run

Check usage of Relational Operator blocks

Not Run

Check usage of Logical Operator blocks

Not Run

Check usage of bit operation blocks

Not Run

Check for blocks not recommended for C/C++ production code deployment

Not Run

Check for inconsistent vector indexing methods

Not Run

Check data types for blocks with index signals

Not Run

Check usage of variant blocks

Not Run

Check usage of lookup table blocks

Not Run

Check usage of Signal Routing blocks

Not Run

Check for root Inports with missing properties

Not Run

Check for root Inports with missing range definitions

Not Run

Check for root Outports with missing range definitions

Not Run

Check usage of Assignment blocks

Not Run

Check global variables in graphical functions

Not Run

Check usage of Gain blocks

Not Run

Check for length of user-defined object names

Not Run

Check data type of loop control variables

Not Run

Stateflow ✓*0* ✗*0* ⚠*0* ▤*14*

Check state machine type of Stateflow charts

Not Run

Check Stateflow charts for ordering of states and transitions

Not Run

Check usage of bitwise operations in Stateflow charts

Not Run

Check for Strong Data Typing with Simulink I/O

Not Run

Check Stateflow debugging options

Not Run

Check Stateflow charts for transition paths that cross parallel state boundaries

Not Run

Check for inappropriate use of transition paths

Not Run

Check Stateflow charts for strong data typing

Not Run

Check naming of ports in Stateflow charts

Not Run

Check scoping of Stateflow data objects

Not Run

Check Stateflow charts for uniquely defined data objects

Not Run

Check usage of shift operations for Stateflow data

Not Run

Check assignment operations in Stateflow charts

Not Run

Check Stateflow charts for unary operators

Not Run

MATLAB ✓*0* ✗*0* ⚠*0* ▤*10*

Check usage of standardized MATLAB function headers

Not Run

Check for MATLAB Function interfaces with inherited properties

Not Run

Check MATLAB Function metrics

Not Run

Check MATLAB Code Analyzer messages

Not Run

Check if/elseif/else patterns in MATLAB Function blocks

Not Run

Check switch statements in MATLAB Function blocks

Not Run

Check usage of relational operators in MATLAB Function blocks

Not Run

Check usage of equality operators in MATLAB Function blocks

Not Run

Check usage of logical operators and functions in MATLAB Function blocks

Not Run

Check type and size of condition expressions

Not Run

Configuration ✓*0* ✗*0* ⚠*0* ☰*32*

Check safety-related diagnostic settings for data store memory

Not Run

Check safety-related diagnostic settings for saving

Not Run

Check safety-related model referencing settings

Not Run

Check safety-related code generation settings for comments

Not Run

Check safety-related code generation interface settings

Not Run

Check safety-related solver settings for simulation time

Not Run

Check safety-related solver settings for solver options

Not Run

Check safety-related solver settings for tasking and sample-time

Not Run

Check safety-related diagnostic settings for solvers

Not Run

Check safety-related diagnostic settings for sample time

Not Run

Check safety-related optimization settings for logic signals

Not Run

Check safety-related block reduction optimization settings

Not Run

Check safety-related code generation settings for code style

Not Run

Check safety-related optimization settings for application lifespan

Not Run

Check safety-related code generation identifier settings

Not Run

Check safety-related optimization settings for loop unrolling threshold

Not Run

Check safety-related optimization settings for data initialization

Not Run

Check safety-related optimization settings for data type conversions

Not Run

Check safety-related optimization settings for division arithmetic exceptions

Not Run

Check safety-related optimization settings for specified minimum and maximum values

Not Run

Check safety-related diagnostic settings for compatibility

Not Run

Check safety-related diagnostic settings for parameters

Not Run

Check safety-related diagnostic settings for Merge blocks

Not Run

Check safety-related diagnostic settings for model initialization

Not Run

Check safety-related diagnostic settings for data used for debugging

Not Run

Check safety-related diagnostic settings for signal connectivity

Not Run

Check safety-related diagnostic settings for bus connectivity

Not Run

Check safety-related diagnostic settings that apply to function-call connectivity

Not Run

Check safety-related diagnostic settings for type conversions

Not Run

Check safety-related diagnostic settings for model referencing

Not Run

Check safety-related diagnostic settings for Stateflow

Not Run

Check safety-related diagnostic settings for signal data

Not Run

Naming ✓*0* ✗*0* ⚠*0* ☰*2*

Check model file name

Not Run

Check model object names

Not Run

Requirements ✓*0* ✗*0* ⚠*0* ▤*1*

▤ Check for model elements that do not link to requirements

Not Run

Code ✓*0* ✗*0* ⚠*0* ▤*2*

▤ Check for blocks not recommended for MISRA C:2012

Not Run

▤ Check configuration parameters for MISRA C:2012

Not Run

Bug Reports ✓0 ✗0 ⚠0 ▤14

▤ *Display bug reports for Embedded Coder*

Not Run

▤ *Display bug reports for IEC Certification Kit*

Not Run

▤ *Display bug reports for Polyspace Code Prover*

Not Run

▤ *Display bug reports for Polyspace Bug Finder*

Not Run

▤ *Display bug reports for Polyspace Code Prover Server*

Not Run

Display bug reports for Polyspace Bug Finder Server

Not Run

Display bug reports for Simulink Design Verifier

Not Run

Display bug reports for Simulink PLC Coder

Not Run

Display bug reports for Simulink Check

Not Run

Display bug reports for Simulink Coverage

Not Run

Display bug reports for Simulink Test

Not Run

Display bug reports for Simulink Requirements

Not Run

Display bug reports for AUTOSAR Blockset

Not Run

Display bug reports for HDL Coder

Not Run

Modeling Standards for IEC 62304 ✓0 ✗0 ⚠0 ☰106

Display configuration management data

Not Run

Display model metrics and complexity report

Not Run

Check for unconnected objects

Not Run

High-Integrity Systems ✓0 ✖0 ⚠0 ▤89

Simulink *✓0 ✖0 ⚠0 ▤28*

Check usage of Abs blocks

Not Run

Check usage of Math Function blocks (rem and reciprocal functions)

Not Run

Check usage of Math Function blocks (log and log10 functions)

Not Run

Check usage of While Iterator blocks

Not Run

Check usage of For and While Iterator subsystems

Not Run

Check usage of For Iterator blocks

Not Run

Check usage of If blocks and If Action Subsystem blocks

Not Run

Check usage of Switch Case blocks and Switch Case Action Subsystem blocks

Not Run

Check usage of conditionally executed subsystems

Not Run

Check usage of Merge blocks

Not Run

Check Relational Operator blocks equating floating-point types

Not Run

Check usage of Relational Operator blocks

Not Run

Check usage of Logical Operator blocks

Not Run

Check usage of bit operation blocks

Not Run

Check for blocks not recommended for C/C++ production code deployment

Not Run

Check for inconsistent vector indexing methods

Not Run

Check data types for blocks with index signals

Not Run

Check usage of variant blocks

Not Run

Check usage of lookup table blocks

Not Run

Check usage of Signal Routing blocks

Not Run

Check for root Inports with missing properties

Not Run

Check for root Inports with missing range definitions

Not Run

Check for root Outports with missing range definitions

Not Run

Check usage of Assignment blocks

Not Run

Check global variables in graphical functions

Not Run

Check usage of Gain blocks

Not Run

Check for length of user-defined object names

Not Run

Check data type of loop control variables

Not Run

Stateflow ✓*0* ✖*0* ⚠*0* ▤*14*

▤ Check state machine type of Stateflow charts

Not Run

▤ Check Stateflow charts for ordering of states and transitions

Not Run

▤ Check usage of bitwise operations in Stateflow charts

Not Run

▤ Check for Strong Data Typing with Simulink I/O

Not Run

▤ Check Stateflow debugging options

Not Run

▤ Check Stateflow charts for transition paths that cross parallel state boundaries

Not Run

▤ Check for inappropriate use of transition paths

Not Run

▤ Check Stateflow charts for strong data typing

Not Run

▤ Check naming of ports in Stateflow charts

Not Run

▤ Check scoping of Stateflow data objects

Not Run

Check Stateflow charts for uniquely defined data objects

Not Run

Check usage of shift operations for Stateflow data

Not Run

Check assignment operations in Stateflow charts

Not Run

Check Stateflow charts for unary operators

Not Run

MATLAB ✓*0* ✗*0* ⚠*0* ☰*10*

Check usage of standardized MATLAB function headers

Not Run

Check for MATLAB Function interfaces with inherited properties

Not Run

Check MATLAB Function metrics

Not Run

Check MATLAB Code Analyzer messages

Not Run

Check if/elseif/else patterns in MATLAB Function blocks

Not Run

Check switch statements in MATLAB Function blocks

Not Run

Check usage of relational operators in MATLAB Function blocks

Not Run

Check usage of equality operators in MATLAB Function blocks

Not Run

Check usage of logical operators and functions in MATLAB Function blocks

Not Run

Check type and size of condition expressions

Not Run

Configuration ✓*0* ✗*0* ⚠*0* ☰*32*

Check safety-related diagnostic settings for data store memory

Not Run

Check safety-related diagnostic settings for saving

Not Run

Check safety-related model referencing settings

Not Run

Check safety-related code generation settings for comments

Not Run

Check safety-related code generation interface settings

Not Run

Check safety-related solver settings for simulation time

Not Run

Check safety-related solver settings for solver options

Not Run

Check safety-related solver settings for tasking and sample-time

Not Run

Check safety-related diagnostic settings for solvers

Not Run

Check safety-related diagnostic settings for sample time

Not Run

Check safety-related optimization settings for logic signals

Not Run

Check safety-related block reduction optimization settings

Not Run

Check safety-related code generation settings for code style

Not Run

Check safety-related optimization settings for application lifespan

Not Run

Check safety-related code generation identifier settings

Not Run

Check safety-related optimization settings for loop unrolling threshold

Not Run

Check safety-related optimization settings for data initialization

Not Run

Check safety-related optimization settings for data type conversions

Not Run

Check safety-related optimization settings for division arithmetic exceptions

Not Run

Check safety-related optimization settings for specified minimum and maximum values

Not Run

Check safety-related diagnostic settings for compatibility

Not Run

Check safety-related diagnostic settings for parameters

Not Run

Check safety-related diagnostic settings for Merge blocks

Not Run

Check safety-related diagnostic settings for model initialization

Not Run

Check safety-related diagnostic settings for data used for debugging

Not Run

Check safety-related diagnostic settings for signal connectivity

Not Run

Check safety-related diagnostic settings for bus connectivity

Not Run

Check safety-related diagnostic settings that apply to function-call connectivity

Not Run

Check safety-related diagnostic settings for type conversions

Not Run

Check safety-related diagnostic settings for model referencing

Not Run

Check safety-related diagnostic settings for Stateflow

Not Run

Check safety-related diagnostic settings for signal data

Not Run

Naming ✓*0* ✗*0* ⚠*0* ☰*2*

Check model file name

Not Run

Check model object names

Not Run

Requirements ✓*0* ✗*0* ⚠*0* ☰*1*

Check for model elements that do not link to requirements

Not Run

Code ✓*0* ✗*0* ⚠*0* ☰*2*

Check for blocks not recommended for MISRA C:2012

Not Run

Check configuration parameters for MISRA C:2012

Not Run

Bug Reports ✓0 ✗0 ⚠0 ▤14

Display bug reports for Embedded Coder

Not Run

Display bug reports for IEC Certification Kit

Not Run

Display bug reports for Polyspace Code Prover

Not Run

Display bug reports for Polyspace Bug Finder

Not Run

Display bug reports for Polyspace Code Prover Server

Not Run

Display bug reports for Polyspace Bug Finder Server

Not Run

Display bug reports for Simulink Design Verifier

Not Run

Display bug reports for Simulink PLC Coder

Not Run

Display bug reports for Simulink Check

Not Run

Display bug reports for Simulink Coverage

Not Run

Display bug reports for Simulink Test

Not Run

Display bug reports for Simulink Requirements

Not Run

Display bug reports for AUTOSAR Blockset

Not Run

Display bug reports for HDL Coder

Not Run

Modeling Standards for ISO 26262 ✓86 ✗0 ⚠20 ▤0

✓ Display configuration management data

Display model configuration and checksum information

Model configuration and checksum information

Attribute	**Value**
Model Version	1.133
Author	ADITYA NIMBALKAR
Date	Wed Jul 15 15:20:37 2020
Model Checksum	3989598269 1814592021 3793947524 394799282

✓ Display model metrics and complexity report

Display number of elements and name, level, and depth of subsystems for the model or subsystem

Model metrics information

Display number of elements for Simulink blocks and Stateflow constructs

Summary

Element Type	Count
Inport	72
Outport	66
SubSystem	34

Simulink

Block Type	Count
Inport	72
Outport	66
SubSystem	34
Constant	17
RelationalOperator	9
Logic	6
Switch	5
Gain	3

Model complexity information

Display name, level, and depth of subsystems

Maximum Subsystem Depth: 5

Subsystem Depth

Subsystem Name	**Level**	**Depth**
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner	1	4
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning	2	2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain1	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain2"	3	1

title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain2		
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain3	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain5" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain5	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain6" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain6	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain7" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain7	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain8" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain8	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData	2	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change	2	2

Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant1	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning	2	2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain1	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain2	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain3	3	1

Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain4" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain4	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain5" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain5	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain6" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain6	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput	2	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	2	3
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional	3	2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional/Slider Gain	4	1
		2

Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning	2	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain1	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain2	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain3	3	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion " title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion	2	1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion " title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion	2	1

∧ Less

✓ Check for unconnected objects

Identify unconnected lines, input ports, and output ports in the model

Passed

There are no unconnected lines, input ports, and output ports in this model.

High-Integrity Systems ✓82 ✗0 ⚠7 ▤0

Simulink ✓*25* ✗*0* ⚠*3* ▤*0*

✓ Check usage of Abs blocks

Identify Abs blocks that have unreachable code or produce overflows

Passed

No Abs blocks found causing unreachable code or produce overflows.

✓ Check usage of Math Function blocks (rem and reciprocal functions)

Identify Math Function blocks using rem and reciprocal functions that cause non-finite results

Passed

All Math Function blocks in the model use operators appropriately.

✓ Check usage of Math Function blocks (log and log10 functions)

Identify Math Function blocks using log and log10 functions that cause non-finite results

Passed

All Math Function blocks in the model use operators appropriately.

✓ Check usage of While Iterator blocks

Identify While Iterator blocks that do not have a positive value for the maximum number of iterations

Passed

No While Iterator blocks found that might cause infinite loops

---

✓ Check usage of For and While Iterator subsystems

Identify sample time-dependent blocks in While and For Iterator subsystems.

Passed

No sample time-dependent blocks in For or While Iterator subsystems.

✓ Check usage of For Iterator blocks

Identify For Iterator blocks that cause variable loops

Passed

No For Iterator blocks found that cause variable loops.

✓ Check usage of If blocks and If Action Subsystem blocks

Identify If and If Action Subsystem blocks without else conditions

Passed

No If blocks with questionable configurations or connections were found.

✓ Check usage of Switch Case blocks and Switch Case Action Subsystem blocks

Identify inappropriately used Switch Case blocks and Switch Case Action Subsystem blocks

Passed

No Switch Case blocks with questionable configurations or connections were found.

✓ Check usage of conditionally executed subsystems

Identify inappropriate blocks in conditionally executed subsystems.

Passed

No blocks with improper sample times or asynchronously executed sample-time dependent blocks were found.

✓ Check usage of Merge blocks

Identify Merge blocks constructs which can lead to ambiguous behavior.

Passed

No merge blocks found which can lead to ambiguous behavior.

✓ Check Relational Operator blocks equating floating-point types

Identify Relational Operator blocks that equate floating-point types

Passed

No Relational Operator blocks found that equate floating-point types.

✓ Check usage of Relational Operator blocks

Identify Relational Operator blocks that operate on different data types or have a non-boolean output

Passed

No Relational Operator blocks found that operate on different data types or have a non-boolean output.

✓ Check usage of Logical Operator blocks

Identify Logical Operator blocks that operate on non-boolean data types

Passed

No Logical Operator blocks found that operate on non-boolean data types.

✓ Check usage of bit operation blocks

Identify bit operation blocks with signed data types as inputs

Passed

No bit operation blocks found with signed data types as inputs.

✓ Check for blocks not recommended for C/C++ production code deployment

Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

Passed

No blocks found which are not recommended for C/C++ production code deployment.

✓ Check for inconsistent vector indexing methods

Identify inconsistent usage of vector indexing methods across the model or subsystem

Passed

No blocks found using inconsistent indexing modes.

✓ Check data types for blocks with index signals

Identify blocks with index signals that have data types other than integers or enums.

Passed

No blocks or charts found with index signals or variables that have data types other than integer or enums.

✓ Check usage of variant blocks

Check variant block settings that might result in code that doesn't trace back to requirements.

Passed

There are no variant blocks that have "Generate preprocessor conditionals" active.

✓ Check usage of lookup table blocks

Check for Lookup Table blocks, Prelookup blocks and Interpolation blocks that do not generate out-of-range checking code.

Passed

No lookup table blocks found to not generate out-of-range checking code.

✓ Check usage of Signal Routing blocks

Identify usage of Signal Routing blocks in Simulink that might impact safety

Passed

No Switch blocks that might generate code with inequality operations (~=) in expressions where at least one side of the expression is a floating-point variable or constant were found.

⚠ Check for root Inports with missing properties

Identify Inport blocks in the top-level of the model with missing or inherited sample times, data types, or port dimensions. Inport block properties are specified with block parameters or Simulink signal data objects that explicitly resolve to the connected signal lines.

Warning

The following Inport blocks have undefined or inherited sample times, data types or port dimensions:

- Wrong_Way_Driver_Warning/ActiveState
- Wrong_Way_Driver_Warning/Current_DataOnOff
- Wrong_Way_Driver_Warning/Data_CountryCode
- Wrong_Way_Driver_Warning/Data_DrivingSide
- Wrong_Way_Driver_Warning/Data_NumberLaneDrivingDirection
- Wrong_Way_Driver_Warning/Data_NumberLaneOppositeDirection
- Wrong_Way_Driver_Warning/Data_OffRoad
- Wrong_Way_Driver_Warning/Data_TurnAngle
- Wrong_Way_Driver_Warning/Diagnostics_MissingData
- Wrong_Way_Driver_Warning/DrivingReverse
- Wrong_Way_Driver_Warning/Sensor_SignAboveRoad
- Wrong_Way_Driver_Warning/Sensor_SignConfidence
- Wrong_Way_Driver_Warning/Sensor_SignID
- Wrong_Way_Driver_Warning/Sensor_SignPositionLateral
- Wrong_Way_Driver_Warning/Sensor_SignPositionLongitudional
- Wrong_Way_Driver_Warning/Sensor_SignRelevance
- Wrong_Way_Driver_Warning/Sensor_SignTrackingState
- Wrong_Way_Driver_Warning/Sensor_SignType
- Wrong_Way_Driver_Warning/Suppressed_Data
- Wrong_Way_Driver_Warning/VehicleSpeed
- Wrong_Way_Driver_Warning/YawRate

∧ Less

Recommended Action

Specify port dimension for the listed Inport blocks or Simulink signal objects.

____ Identify Inport blocks in the top-level of the model with missing or inherited sample times, data types, or port dimensions. Inport block properties are specified with block parameters or Simulink signal data objects that explicitly resolve to the connected signal lines.

Warning

The following Inport blocks have undefined or inherited sample times, data types or port dimensions:

- Wrong_Way_Driver_Warning/ActiveState
- Wrong_Way_Driver_Warning/Current_DataOnOff
- Wrong_Way_Driver_Warning/Data_CountryCode
- Wrong_Way_Driver_Warning/Data_DrivingSide
- Wrong_Way_Driver_Warning/Data_NumberLaneDrivingDirection
- Wrong_Way_Driver_Warning/Data_NumberLaneOppositeDirection
- Wrong_Way_Driver_Warning/Data_OffRoad
- Wrong_Way_Driver_Warning/Data_TurnAngle
- Wrong_Way_Driver_Warning/Diagnostics_MissingData
- Wrong_Way_Driver_Warning/DrivingReverse
- Wrong_Way_Driver_Warning/Sensor_SignAboveRoad
- Wrong_Way_Driver_Warning/Sensor_SignConfidence
- Wrong_Way_Driver_Warning/Sensor_SignID
- Wrong_Way_Driver_Warning/Sensor_SignPositionLateral
- Wrong_Way_Driver_Warning/Sensor_SignPositionLongitudional
- Wrong_Way_Driver_Warning/Sensor_SignRelevance
- Wrong_Way_Driver_Warning/Sensor_SignTrackingState
- Wrong_Way_Driver_Warning/Sensor_SignType
- Wrong_Way_Driver_Warning/Suppressed_Data
- Wrong_Way_Driver_Warning/VehicleSpeed

- Wrong_Way_Driver_Warning/YawRate

∧ Less

Recommended Action

Specify sample time information for the listed Inport blocks or Simulink signal objects. Note: The sample time of root Inports with bus type must match the sample times specified at the leaf elements of the bus object.

✓ Check for root Inports with missing range definitions

Identify root-level Inport blocks with missing or erroneous minimum or maximum values. Inport block minimum and maximum values are specified with block parameters or Simulink signal objects that explicitly resolve to the connected signal lines.

Passed

There are no missing or erroneous Inport range properties at the model root level.

✓ Check for root Outports with missing range definitions

Identify root-level Outport blocks with missing or erroneous minimum or maximum values. Outport block minimum and maximum values are specified with block parameters or Simulink signal objects that explicitly resolve to the connected signal lines.

Passed

There are no missing or erroneous Outport range properties at the model root level.
Note: Root Outports with inherited data types are not analyzed by this check.

✓ Check usage of Assignment blocks

Identify Assignment blocks whose array fields are not initialized.

Passed

All Assignment blocks are configured with block parameter "Action if any output element is not assigned" set to Warning or Error.

✓ Check global variables in graphical functions

Identify expressions that both read and write to the same global data.

Passed

No expressions found that both read and write to the same global data.

⚠ Check usage of Gain blocks

Identify Gain blocks with value which resolves to 1

Warning

The following Gain blocks have value which resolves to 1.

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

-
-

∧ Less

Recommended Action

Consider remodeling to remove the Gain blocks with values that resolve to 1

⚠ Check for length of user-defined object names

Identify user-defined object names with length greater than threshold

Warning

The following data objects have name length greater than threshold (31).

Data Objects	Source
Data_NumberLaneOppositeDirection	WWDW_dd.sldd
Sensor_SignPositionLongitudional	WWDW_dd.sldd

Recommended Action

Change the names of mentioned data objects to have length less than 31.

✓ Check data type of loop control variables

Identify loop control variables using non-integer data types.

Passed

No For Iterator blocks or MATLAB Function blocks found using non-integer data type for loop control counter variable.

Stateflow ✓14 ☒0 ⚠0 ▤0

✓ Check state machine type of Stateflow charts

Identify Stateflow Charts whose State Machine Type differs from the type set in the Model Advisor Configuration Editor.

Passed

No Stateflow Charts found that deviate from recommended state machine type.

✓ Check Stateflow charts for ordering of states and transitions

Identify Stateflow charts that do not use explicit ordering of parallel states and transitions.

Passed

No Stateflow Charts found that deviate from recommended state/transition execution order settings.

✓ Check usage of bitwise operations in Stateflow charts

Identify usage of signed data type operands to bitwise operations in Stateflow charts.

Passed

No Stateflow objects found that use signed data type operands with bitwise operations.

✓ Check for Strong Data Typing with Simulink I/O

Verify configuration settings for strong data typing on the boundaries between Simulink and Stateflow

Passed

No Stateflow charts found that set 'Use Strong Data Typing with Simulink I/O' to off.

✓ Check Stateflow debugging options

Identify whether Stateflow debugging options are set appropriately

Passed

All Stateflow debugging options are set appropriately.

✓ Check Stateflow charts for transition paths that cross parallel state boundaries

Identify transition paths that cross parallel state boundaries in Stateflow charts.

Passed

No transition paths crossing parallel state boundaries were found in Stateflow charts.

✓ Check for inappropriate use of transition paths

Identify transition paths that go into and out of a state without ending on a substate.

Passed

No transition paths found that go into and out of a state without ending on a substate.

✓ Check Stateflow charts for strong data typing

Identify expressions with variables and parameters of different data types in Stateflow objects.

Passed

No expressions were found with variables and parameters of different data types.

✓ Check naming of ports in Stateflow charts

Identify mismatches between names of Stateflow ports and associated signals

Passed

There are no name mismatches between Stateflow ports and associated signals

---

✓ Check scoping of Stateflow data objects

Identify Stateflow data objects with local scope that are not scoped at the chart level or below

Passed

All Stateflow data objects are properly scoped.

✓ Check Stateflow charts for uniquely defined data objects

Identify local data identifiers that are defined in multiple scopes within a chart.

Passed

No Stateflow data identifiers found to be defined in multiple scopes.

✓ Check usage of shift operations for Stateflow data

Identify usage of Stateflow bit-shifting operations that might impact safety.

Passed

There are no Stateflow bit-shifting operations greater than the bit-width of the input or output type.

✓ Check assignment operations in Stateflow charts

Identify assignment operations in Stateflow objects which cast integer and fixed-point calculations to wider datatype.

Passed

No assignment operations were found which cast integer and fixed-point calculations to wider datatype.

✓ Check Stateflow charts for unary operators

Identify unary minus operators on unsigned data types in Stateflow objects.

Passed

No unary minus operations on unsigned data types were found in Stateflow objects.

MATLAB ✓*10* ✗*0* ⚠*0* ☰*0*

✓ Check usage of standardized MATLAB function headers

Identify usage of standardized function headers in MATLAB function.

Passed

No MATLAB function blocks found without standardized function headers.

✓ Check for MATLAB Function interfaces with inherited properties

Identify MATLAB Functions that have inputs, outputs, or parameters with inherited complexity or data type properties.

Passed

No MATLAB Function interfaces with inherited complexity or data type properties found.

✓ Check MATLAB Function metrics

Identify MATLAB Functions that violate code and complexity metrics.

Passed

No MATLAB Function blocks found that violate code and complexity metrics.

✓ Check MATLAB Code Analyzer messages

Check MATLAB functions for %#codegen directive, MATLAB Code Analyzer messages, and justification message IDs.

Passed

No MATLAB Function blocks found with Code Analyzer messages, missing %#codegen directive or inappropriate usage of justification message IDs.

✓ Check if/elseif/else patterns in MATLAB Function blocks

Identify if/elseif/else patterns without appropriate else conditions in embedded MATLAB code

Passed

No inappropriate if/elseif/else patterns found.

✓ Check switch statements in MATLAB Function blocks

Identify inappropriately used switch statements in embedded MATLAB code

Passed

No inappropriately used switch statements found.

✓ Check usage of relational operators in MATLAB Function blocks

Identify relational operators operating on operands of different data types in MATLAB Function blocks.

Passed

No relational operators found operating on operands of different data types.

✓ Check usage of equality operators in MATLAB Function blocks

Identify equality operators used with floating-point operands in MATLAB Function blocks.

Passed

No equality operators found operating on floating-point operands.

✓ Check usage of logical operators and functions in MATLAB Function blocks

Identify logical operators and functions operating on operands with numerical data types.

Passed

No logical operators or functions found operating on operands with numerical data types.

✓ Check type and size of condition expressions

Identify condition expressions which are not logical scalars.

Passed

No condition expressions found which are not logical scalars.

Configuration ✓*30* ✖*0* ⚠*2* ☰*0*

✓ Check safety-related diagnostic settings for data store memory

Check diagnostic settings in the model configuration that apply to data store memory and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	**Parameter**	**Current Value**	**Recommended Values**
Pass	Detect read before write (ReadBeforeWriteMsg)	EnableAllAsError	EnableAllAsError
Pass	Detect write after read (WriteAfterReadMsg)	EnableAllAsError	EnableAllAsError
Pass	Detect write after write (WriteAfterWriteMsg)	EnableAllAsError	EnableAllAsError

Pass	Multitask data store (MultiTaskDSMMsg)	error	error
Pass	Duplicate data store names (UniqueDataStoreMsg)	error	error

✓ Check safety-related diagnostic settings for saving

Check diagnostic settings in the model configuration that apply to saving model files.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Block diagram contains disabled library links (SaveWithDisabledLinksMsg)	error	error
Pass	Block diagram contains parameterized library links (SaveWithParameterizedLinksMsg)	error	error

✓ Check safety-related model referencing settings

Check model referencing settings in the model configuration that might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Rebuild (UpdateModelReferenceTargets)	IfOutOfDateOrStructuralChange	AssumeUpToDate, IfOutOfDateOrStructuralChange
Pass	Pass fixed-size scalar root inputs by value for code generation (ModelReferencePassRootInputsByReference) *	on	on
Pass	Minimize algebraic loop occurrences (ModelReferenceMinAlgLoopOccurrences)	off	off

Recommended Action

* The Command-Line values provided in the table are reverse of the settings in the Configuration Parameters Dialog. Therefore, 'on' in the Command-Line corresponds to an "Off" setting in the dialog, and 'off' in the Command-Line corresponds to an "On" setting in the dialog.

✓ Check safety-related code generation settings for comments

Check code generation settings in the model configuration that apply comments and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Include comments (GenerateComments)	on	on	
Pass	Simulink block comments (SimulinkBlockComments)	on	on	GenerateComments
Pass	Show eliminated blocks (ShowEliminatedStatement)	on	on	GenerateComments
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	
Pass	Verbose comments for 'Model default' storage class (ForceParamTrailComments)	on	on	GenerateComments
D - Pass	Include comments (GenerateComments)	on	on	
Pass	Requirements in block comments (ReqsInCode)	on	on	SystemTargetFile, GenerateComments

---

✓ Check safety-related code generation interface settings

Check code generation interface settings in the model configuration that might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites

Pass	non-finite numbers (SupportNonFinite)	off	off	
Pass	absolute time (SupportAbsoluteTime)	off	off	SystemTargetFile
Pass	continuous time (SupportContinuousTime)	off	off	SystemTargetFile
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	
Pass	non-inlined S-functions (SupportNonInlinedSFcns)	off	off	SystemTargetFile
Pass	Classic call interface (GRTInterface)	off	off	
Pass	Single output/update function (CombineOutputUpdateFcns)	on	on	
Pass	Terminate function required (IncludeMdlTerminateFcn)	off	off	SystemTargetFile
Pass	Remove error status field in real-time model data structure (SuppressErrorStatus)	on	on	SystemTargetFile
Pass	MAT-file logging (MatFileLogging)	off	off	

✓ Check safety-related solver settings for simulation time

Identify if the model Start time is set to 0 and Stop time is less than the Application Life Span.

Passed

No issues found with solver settings for simulation time.

✓ Check safety-related solver settings for solver options

Check solver settings in the model configuration that apply to solvers and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Type (SolverType)	Fixed-step	Fixed-step
Pass	Solver (SolverName)	FixedStepDiscrete	FixedStepDiscrete

⚠ Check safety-related solver settings for tasking and sample-time

Check solver settings in the model configuration that apply to tasking and sample-time constraints and might impact safety.

Warning

The model configuration parameters are not set to the recommended values specified in the data file.

Status	Parameter	Current Value	Not Recommended Values
Warning	Automatically handle rate transition for data transfer (AutoInsertRateTranBlk)	on	on

Recommended Action

Clear **Automatically handle rate transition for data transfer** checkbox.

✓ Check safety-related diagnostic settings for solvers

Check diagnostic settings in the model configuration that apply to solvers and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Algebraic loop (AlgebraicLoopMsg)	error	error
Pass	Minimize algebraic loop (ArtificialAlgebraicLoopMsg)	error	error
Pass	Block priority violation (BlockPriorityViolationMsg)	error	error
Pass	Automatic solver parameter selection (SolverPrmCheckMsg)	error	error
Pass	State name clash (StateNameClashWarn)	warning	warning

✓ Check safety-related diagnostic settings for sample time

Check diagnostic settings in the model configuration that apply to sample time and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Source block specifies -1 sample time (InheritedTsInSrcMsg)	error	error

Pass	Multitask rate transition (MultiTaskRateTransMsg)	error	error
Pass	Multitask conditionally executed subsystem (MultiTaskCondExecSysMsg)	error	error
Pass	Enforce sample times specified by Signal Specification blocks (SigSpecEnsureSampleTimeMsg)	error	error
Pass	Single task rate transition (SingleTaskRateTransMsg)	error	error
Pass	Tasks with equal priority (TasksWithSamePriorityMsg)	error	error
Pass	Unspecified inheritability of sample time (UnknownTsInhSupMsg)	error	error

---

✓ Check safety-related optimization settings for logic signals

Check optimization settings in the model configuration that apply to logic signals and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Implement logic signals as Boolean data (vs. double) (BooleanDataType)	on	on

---

✓ Check safety-related block reduction optimization settings

Check block reduction optimization settings in the model configuration that might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Block reduction (BlockReduction)	off	off

✓ Check safety-related code generation settings for code style

Check code generation settings in the model configuration that apply to code style and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Parentheses level (ParenthesesLevel)	Maximum	Maximum	SystemTargetFile
Pass	Preserve operand order in expression (PreserveExpressionOrder)	on	on	SystemTargetFile
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	

✓ Check safety-related optimization settings for application lifespan

Check optimization settings in the model configuration that apply to application lifespan and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Application lifespan (days) (LifeSpan)	inf	Inf

✓ Check safety-related code generation identifier settings

Check code generation identifier settings in the model configuration that might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Not Recommended Values	Prerequisites
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target		
Pass	Minimum mangle length (MangleLength)	4		1, 2, 3	SystemTargetFile

---

✓ Check safety-related optimization settings for loop unrolling threshold

Check optimization settings in the model configuration that apply to loop unrolling threshold and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Not Recommended Values
Pass	Loop unrolling threshold (RollThreshold)	5	0, 1

⚠ Check safety-related optimization settings for data initialization

Check optimization settings in the model configuration that apply to data initialization and might impact safety.

Warning

The model configuration parameters are not set to the recommended values specified in the data file.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Warning	Remove root level I/O zero initialization (ZeroExternalMemoryAtStartup) *	off	on	SystemTargetFile

Warning	Remove internal data zero initialization (ZeroInternalMemoryAtStartup) *	off	on	SystemTargetFile

Recommended Action

Follow the links in the result table to modify the model configuration parameters.

* The Command-Line values provided in the table are reverse of the settings in the Configuration Parameters Dialog. Therefore, 'on' in the Command-Line corresponds to an "Off" setting in the dialog, and 'off' in the Command-Line corresponds to an "On" setting in the dialog.

✓ Check safety-related optimization settings for data type conversions

Check optimization settings in the model configuration that apply to data type conversions and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Remove code from floating-point to integer conversions that wraps out-of-range values (EfficientFloat2IntCast)	on	on

---

✓ Check safety-related optimization settings for division arithmetic exceptions

Check optimization settings in the model configuration that apply to division arithmetic exceptions and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	
Pass	Remove code that protects against division arithmetic exceptions (NoFixptDivByZeroProtection)	off	off	SystemTargetFile

---

✓ Check safety-related optimization settings for specified minimum and maximum values

Check optimization settings in the model configuration that apply to specified minimum and maximum values and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites

Pass	Optimize using the specified minimum and maximum values (UseSpecifiedMinMax)	off	off	SystemTargetFile
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	

✓ Check safety-related diagnostic settings for compatibility

Check diagnostic settings in the model configuration that affect compatibility and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	S-function upgrades needed (SFcnCompatibilityMsg)	error	error

---

✓ Check safety-related diagnostic settings for parameters

Check diagnostic settings in the model configuration that apply to parameters and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Detect downcast (ParameterDowncastMsg)	error	error
Pass	Detect underflow (ParameterUnderflowMsg)	error	error
Pass	Detect overflow (ParameterOverflowMsg)	error	error
Pass	Detect precision loss (ParameterPrecisionLossMsg)	error	error
Pass	Detect loss of tunability (ParameterTunabilityLossMsg)	error	error

✓ Check safety-related diagnostic settings for Merge blocks

Check diagnostic settings in the model configuration that apply to Merge blocks and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Detect multiple driving blocks executing at the same time step (MergeDetectMultiDrivingBlocksExec)	error	error

✓ Check safety-related diagnostic settings for model initialization

Check diagnostic settings in the model configuration that affect model initialization and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Underspecified initialization detection (UnderspecifiedInitializationDetection)	Simplified	Simplified

---

✓ Check safety-related diagnostic settings for data used for debugging

Check diagnostic settings in the model configuration that apply to data used for debugging and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Model Verification block enabling (AssertControl)	DisableAll	DisableAll

---

✓ Check safety-related diagnostic settings for signal connectivity

Check diagnostic settings in the model configuration that apply to signal connectivity and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Signal label mismatch (SignalLabelMismatchMsg)	error	error
Pass	Unconnected block input ports (UnconnectedInputMsg)	error	error
Pass	Unconnected block output ports (UnconnectedOutputMsg)	error	error
Pass	Unconnected line (UnconnectedLineMsg)	error	error

✓ Check safety-related diagnostic settings for bus connectivity

Check diagnostic settings in the model configuration that apply to bus connectivity and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unspecified bus object at root Outport block (RootOutportRequireBusObject)	error	error
Pass	Element name mismatch	error	error

	(BusObjectLabelMismatch)		
Pass	Bus signal treated as vector (StrictBusMsg)	ErrorOnBusTreatedAsVector	ErrorOnBusTreatedAsVector
Pass	Non-bus signals treated as bus signals (NonBusSignalsTreatedAsBus)	error	error

---

✓ Check safety-related diagnostic settings that apply to function-call connectivity

Check diagnostic settings in the model configuration that apply to function-call connectivity and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	InvalidFcnCallConnMsg	error	error
Pass	Context-dependent inputs (FcnCallInpInsideContextMsg)	error	error

---

✓ Check safety-related diagnostic settings for type conversions

Check diagnostic settings in the model configuration that apply to type conversions and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unnecessary type conversions (UnnecessaryDatatypeConvMsg)	warning	warning
Pass	Vector/matrix block input conversion (VectorMatrixConversionMsg)	error	error
Pass	32-bit integer to single precision float conversion (Int32ToFloatConvMsg)	warning	warning

---

✓ Check safety-related diagnostic settings for model referencing

Check diagnostic settings in the model configuration that apply to model referencing and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Port and parameter mismatch (ModelReferenceIOMismatchMessage)	error	error
Pass	Invalid root Inport/Outport block connection (ModelReferenceIOMsg)	error	error
Pass	Unsupported data logging (ModelReferenceDataLoggingMessage)	error	error

✓ Check safety-related diagnostic settings for Stateflow

Check diagnostic settings in the model configuration that apply to Stateflow and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unexpected backtracking (SFUnexpectedBacktrackingDiag)	error	error
Pass	Invalid input data access in chart initialization (SFInvalidInputDataAccessInChartInitDiag)	error	error
Pass	No unconditional default transitions (SFNoUnconditionalDefaultTransitionDiag)	error	error
Pass	Transition outside natural parent (SFTransitionOutsideNaturalParentDiag)	error	error
Pass	Unreachable execution path (SFUnreachableExecutionPathDiag)	error	error
Pass	Undirected event broadcasts (SFUndirectedBroadcastEventsDiag)	error	error
Pass	Transition action specified before condition action (SFTransitionActionBeforeConditionDiag)	error	error

✓ Check safety-related diagnostic settings for signal data

Check diagnostic settings in the model configuration that apply to signal data and might impact safety.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Signal resolution (SignalResolutionControl)	UseLocalSettings	None, UseLocalSettings
Pass	Division by singular matrix (CheckMatrixSingularityMsg)	error	error
Pass	Underspecified data types (UnderSpecifiedDataTypeMsg)	error	error
Pass	Wrap on overflow (IntegerOverflowMsg)	error	error
Pass	Saturate on overflow (IntegerSaturationMsg)	error	error
Pass	Inf or NaN block output (SignalInfNanChecking)	error	error
Pass	"rt" prefix for identifiers (RTPrefix)	error	error
Pass	Simulation range checking (SignalRangeChecking)	error	error

📁 *Naming* ✓*1* ✗*0* ⚠*1* ▤*0*

✓ Check model file name

Identify inappropriate characters and length issues in model file name

Passed

No issues found with model file name.

⚠ Check model object names

Identify invalid names of following model objects (first invalid name fragment is highlighted):

- Blocks
- Signals
- Parameters
- Buses
- Stateflow elements

Warning

The following model objects have invalid names:

Block	Name
Wrong_Way_Driver_Warning/Data_NumberLaneOppositeDirection" title="Wrong_Way_Driver_Warning/Data_NumberLaneOppositeDirection	
Wrong_Way_Driver_Warning/Sensor_SignPositionLongitudional" title="Wrong_Way_Driver_Warning/Sensor_SignPositionLongitudional	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Data_NumberLaneOppositeDirection" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Data_NumberLaneOppositeDirection	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Sensor_SignPositionLongitudional" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Sensor_SignPositionLongitudional	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Sensor_SignPositionLongitudional" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Sensor_SignPositionLongitudional	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain	SliderGain
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Cond	SliderGain1

itioning/Slider Gain1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain1	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain2	SliderGain2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain3	SliderGain3
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain5" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain5	SliderGain5
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain6" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain6	SliderGain6
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain7" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain7	SliderGain7
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain8" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning/Slider Gain8	SliderGain8

Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Relational Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Relational Operator	RelationalOperator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant	CompareToConstant
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Compare To Constant1	CompareToConstant1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Logical Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Logical Operator	LogicalOperator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Logical Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Logical Operator1	LogicalOperator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Logical Operator2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Logical Operator2	LogicalOperator2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Relational Operator"	RelationalOperator

title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Relational Operator	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Relational Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Relational Operator1	RelationalOperator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Relational Operator2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Relational Operator2	RelationalOperator2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Data_NumberLaneOppositeDirection" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Data_NumberLaneOppositeDirection	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain	SliderGain
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain1	SliderGain1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain2	SliderGain2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider	SliderGain3

Gain3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain3	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain4" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain4	SliderGain4
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain5" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain5	SliderGain5
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain6" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning/Slider Gain6	SliderGain6
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Relational Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Relational Operator	RelationalOperator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator	LogicalOperator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/L	LogicalOperator1

ogical Operator1	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator	RelationalOperator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator1	RelationalOperator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator2	RelationalOperator2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator3	RelationalOperator3
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional/Slider Gain	SliderGain
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain	SliderGain
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider	SliderGain1

Gain1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain1	
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain2	SliderGain2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning/Slider Gain3	SliderGain3
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion " title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion	Vehicle_Speed_Conversion
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion " title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion	Yaw_Rate_Conversion

∧ Less

Signal	Name
Wrong_Way_Driver_Warning	
Wrong_Way_Driver_Warning	

Parameter used in	Name	Defined in

Wrong_Way_Driver_Warning/Data_NumberLaneOppositeDirection		data dictionary
Wrong_Way_Driver_Warning/Sensor_SignPositionLongitudional		data dictionary

Recommended Action

Change flagged names of model objects

Requirements ✓*0* ✗*0* ⚠*1* ▤*0*

⚠ Check for model elements that do not link to requirements

Check for model elements that do not link to a requirements document.

Warning

The following model elements do not link to a requirements document:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData
-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
-
-
-

Recommended Action

For each block in the list, in the Model Editor, right-click the block, select `Requirements`, and specify a requirement.

Code ✓*2* ✗*0* ⚠*0* ▤*0*

✓ Check for blocks not recommended for MISRA C:2012

Identify blocks that are not recommended for MISRA C:2012 compliant code generation.

Passed

None of the blocks are defined as "not recommended" for MISRA C:2012 compliant code generation.

✓ Check configuration parameters for MISRA C:2012

Identify configuration parameters that might impact MISRA C:2012 compliant code generation.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Model Verification block enabling (AssertControl)	DisableAll	DisableAll	
D - Pass	Shared code placement (UtilityFuncGeneration)	Shared location	Shared location	
Pass	Generate shared constants (GenerateSharedConstants)	off	off	UtilityFuncGeneration
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	
Pas	continuous time (SupportContinuousTim	off	off	SystemTarget

s	e)			File
Pass	non-inlined S-functions (SupportNonInlinedSFcns)	off	off	SystemTarget File
Pass	MAT-file logging (MatFileLogging)	off	off	
Pass	Code replacement library (CodeReplacementLibrary)	None	None, AUTOSAR 4.0	
Pass	Parentheses level (ParenthesesLevel)	Maximum	Maximum	SystemTarget File
Pass	Casting modes (CastingMode)	Standards	Standards	SystemTarget File
Pass	System-generated identifiers (InternalIdentifier)	Shortened	Shortened	SystemTarget File
Pass	Signed integer division rounds to (ProdIntDivRoundTo)	Zero	Zero, Floor	
Pass	Use division for fixed-point net slope computation (UseDivisionForNetSlopeComputation)	UseDivisionForReciprocalsOfIntegersOnly	on, UseDivisionForReciprocalsOfIntegersOnly	
Pass	Replace multiplications by powers of two with signed bitwise shifts (EnableSignedLeftShifts)	off	off	SystemTarget File
Pass	Allow right shifts on signed integers (EnableSignedRightShifts)	off	off	SystemTarget File
	Wrap on overflow	error	warning, error	

Pass	(IntegerOverflowMsg)			
Pass	Inf or NaN block output (SignalInfNanChecking)	error	warning, error	
Pass	Dynamic memory allocation in MATLAB functions (MATLABDynamicMemAlloc)	off	off	
Pass	External mode (ExtMode)	off	off	
Pass	Undirected event broadcasts (SFUndirectedBroadcastEventsDiag)	error	error	
Pass	Compile-time recursion limit for MATLAB functions (CompileTimeRecursionLimit)	0	0	
Pass	Enable run-time recursion for MATLAB functions (EnableRuntimeRecursion)	off	off	
D - Pass	Include comments (GenerateComments)	on	on	
Pass	MATLAB user comments (MATLABFcnDesc)	on	on	GenerateComments, SystemTargetFile

∧ Less

Bug Reports ✓1 ✗0 ⚠13 ☰0

⚠ *Display bug reports for Embedded Coder*

Display bug reports for Embedded Coder (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 49 Embedded Coder bug reports for release R2020a

ID	Bug Report Summary	Modified
2248226	Embedded Coder - **Incorrect Code Generation**: Incorrect results might occur for a model containing a Data Store Memory block interacting with a reusable subsystem configured to minimize algebraic loop occurrences	13 Jul 2020
2029502	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model containing multiple Reusable custom storage class with a branched root Inport	10 Jul 2020
2248045	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model containing Data Store Memory block and a For Iterator Subsystem or a While Iterator Subsystem	10 Jul 2020
2247270	Embedded Coder - **Incorrect Code Generation**: Incorrect initial value for block output inside reusable subsystem	10 Jul 2020
2221375	Embedded Coder - **Incorrect Code Generation**: Numerical mismatch between normal and accelerator mode simulation for variable dimension inputs when the configuration parameter UseRowMajorAlgorithm is selected	10 Jul 2020
2204585	Embedded Coder - **Incorrect Code Generation**: Signal object InitialValue ignored on root inputs of referenced models when storage class is 'Model	10 Jul 2020

	default'	
2249030	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model with Reset Function block	10 Jul 2020
2119697	Embedded Coder - Stateflow chart inside rate grouped Simulink Function might lead to assertion during code generation	09 Jul 2020
2176178	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model containing a bus data type across a reusable atomic subsystem	26 Jun 2020
2203079	Embedded Coder - Uncompilable generated code might occur for MATLAB code containing a loop that operates on variables of different data types with SIMD enabled	11 Jun 2020
2244678	Embedded Coder - **Incorrect Code Generation**: Incorrect code might be generated for a model containing a MATLAB Function block with similar expressions over struct type variables	11 Jun 2020
2238014	Embedded Coder - Error when calling TLC library function LibBlockInputSignalAllowScalarExpandedExpr	11 Jun 2020
2232273	Embedded Coder - **Incorrect Code Generation**: Constant sample time output signal in referenced model might lead to incorrect code	11 Jun 2020
2221392	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model with a Unit Delay block inside a For Iterator or While Iterator subsystem	11 Jun 2020
2218634	Embedded Coder - Missing example files in documentation topic "Access Data Through Functions by Using Storage Classes in Embedded Coder Dictionary"	11 Jun 2020
2210185	Embedded Coder - AUTOSAR Diagnostic Event Manager event failure not flagged if event ID counter exceeds rather than meets failure threshold	11 Jun 2020
2207911	Embedded Coder - **Incorrect Code Generation**: Incorrect generated code for model with DQ Limiter block and Inverse Park Transform block	02 Jun 2020
2122070	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model with Unit Delay block inside a For Each Subsystem block	29 May 2020
2192341	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model with Stateflow chart	28 May 2020
2216985	Embedded Coder - **Incorrect Code Generation**: Incorrect code generation for a function-call, triggered, or enabled and triggered subsystem that is configured for reusable function packaging	26 May 2020
2111370	Embedded Coder - Persistent global variable used within a Parallel for-Loops(parfor) present in a MATLAB Function block or MATLAB System block may result in code that does not compile	14 May 2020

2215349	Embedded Coder - MATLAB may crash when using the getDataInterfaces function of Code Descriptor API	14 May 2020
2213080	Embedded Coder - MATLAB might crash when generating code for a model that contains subsystems	14 May 2020
2209352	Embedded Coder - Code generation error with global Data Store Memory and Export-Function model	14 May 2020
2191117	Embedded Coder - **Incorrect Code Generation**: Tunable parameters in non-inlined S-function might lead to incorrect code	14 May 2020
2189985	Embedded Coder - **Incorrect Code Generation**: Incorrect initial value for block output inside reusable subsystem	14 May 2020
2181053	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model containing a series of directly connected Bus Creator blocks	14 May 2020
2178595	Embedded Coder - SIL simulation with Microsoft Visual C++ compiler option /TP produces compiler error	14 May 2020
2166906	Embedded Coder - SIL/PIL simulation fails if model contains Reset Function block and model step function uses function prototype control	14 May 2020
2218742	Embedded Coder - **Incorrect Code Generation**: Generated code does not initialize instance-specific parameters for models that specify dynamic allocation	14 May 2020
2199240	Embedded Coder - Code generation error when subsystem contains Stateflow chart and execution time profiling is enabled	14 May 2020
2190021	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model containing a Selector block that connects to a Unit Delay block	14 May 2020
2192241	Embedded Coder - XCP-based external mode fails for binaries with debug symbols for empty compilation units	23 Apr 2020
2190935	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model containing Data Store Memory and MATLAB System blocks	22 Apr 2020
2197821	MATLAB Coder - **Incorrect Code Generation**: Output of set operations with the 'rows' option might not be in sortrows order when NaNs are present	07 Apr 2020
2194951	Embedded Coder - Performance regression caused during code generation for models with large data set	06 Apr 2020
2176228	Embedded Coder - Embedded Coder fails to generate correct code from a Simulink Code Inspector compatible model if it defines instance parameters	30 Mar 2020
2192558	Embedded Coder - **Incorrect Code Generation**: Customized step function prototype with custom storage class on Root-level Outport might generate	30 Mar 2020

	incorrect code	
2133942	Embedded Coder - Code generator places code for asynchronously triggered atomic subsystem in wrong location	19 Mar 2020
1934700	Embedded Coder - Model block SIL or PIL simulation produces error for AUTOSAR software component with model workspace parameters mapped to SharedParameter	06 Feb 2020
2133775	Embedded Coder - MATLAB might crash when generating code for a model containing C action language Stateflow Chart with shift operation applied to custom storage class	06 Feb 2020
2131505	Embedded Coder - **Incorrect Code Generation**: Model that uses row-major array layout and complex types containing fixed-point data types might generate incorrect results	30 Jan 2020
2164963	Embedded Coder - Non-deterministic ordering of Rte_Read API calls for SenderReceiver ports in generated AUTOSAR code	29 Jan 2020
2106435	Embedded Coder - Code generation error for AUTOSAR model in which Simulink Function sends message to root outport	19 Dec 2019
2072645	Embedded Coder - **Incorrect Code Generation**: Incorrect results might occur for C++ std::string in MATLAB Function block	11 Oct 2019
1999672	Embedded Coder - **Incorrect Code Generation**: Incorrect results are possible for a model with a For Each subsystem block	16 Aug 2019
2007592	Embedded Coder - **Incorrect Code Generation**: Incorrect results might occur for a model with a Bus Assignment block and an Assignment block	16 Aug 2019
1955846	Embedded Coder - MATLAB might crash while building a model with a Reusable custom storage class specification on root i/o	24 Apr 2019
1709275	Embedded Coder - Generated code for Stateflow Chart may contain dead initialization code	12 Feb 2018

∧ Less

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

✓ *Display bug reports for IEC Certification Kit*

Display bug reports for IEC Certification Kit (R2020a) available at https://www.mathworks.com/support/bugreports .
NOTE: This check does not determine whether your model might be affected by these bugs.

Passed

There are no IEC Certification Kit bug reports for release R2020a.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Polyspace Code Prover*

Display bug reports for Polyspace Code Prover (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 5 Polyspace Code Prover bug reports for release R2020a

ID	Bug Report Summary	Modified
2234024	Polyspace Code Prover - Error with behavior specification options in Polyspace analysis in client-server mode	01 Jul 2020

1654557	Polyspace Code Prover - Operation using wrapped values from a previous orange overflow is green even if tooltip indicates a possible second overflow	15 Jun 2020
2142882	Polyspace Code Prover - External constraints are not recognized on arguments passed by reference to stubbed functions	22 Apr 2020
2190091	Polyspace Code Prover - Error during compilation of C++ file: stl_tree.h, line 2142: error: no instance of constructor	22 Apr 2020
2184422	Polyspace Code Prover - Incorrect Function not called check when using compiler pragma inline=never	12 Mar 2020

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	**Value**
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Polyspace Bug Finder*

Display bug reports for Polyspace Bug Finder (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 2 Polyspace Bug Finder bug reports for release R2020a

ID	Bug Report Summary	Modified
2198724	Polyspace Bug Finder - Launching an analysis from MATLAB generates the error: Product required for 'pslinkrunImpl' not installed	14 May 2020
2211362	Polyspace Bug Finder - Polyspace analysis fails with error about anonymous union members	13 May 2020

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Polyspace Code Prover Server*

Display bug reports for Polyspace Code Prover Server (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 2 Polyspace Code Prover Server bug reports for release R2020a

ID	Bug Report Summary	Modified
	Polyspace Code Prover - Operation using wrapped values from a previous	15 Jun

1654557	orange overflow is green even if tooltip indicates a possible second overflow	2020
2184422	Polyspace Code Prover - Incorrect Function not called check when using compiler pragma inline=never	12 Mar 2020

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Polyspace Bug Finder Server*

Display bug reports for Polyspace Bug Finder Server (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 2 Polyspace Bug Finder Server bug reports for release R2020a

ID	Bug Report Summary	Modified
2198724	Polyspace Bug Finder - Launching an analysis from MATLAB generates the error: Product required for 'pslinkrunImpl' not installed	14 May 2020
	Polyspace Bug Finder - Polyspace analysis fails with error about anonymous	13 May

2211362	union members	2020

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Simulink Design Verifier*

Display bug reports for Simulink Design Verifier (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 12 Simulink Design Verifier bug reports for release R2020a

ID	Bug Report Summary	Modified
1796913	Simulink Design Verifier - Incorrect Code Generation: Incorrect dead logic reported for multiport switch having constant array as control input	10 Jul 2020
2172875	Simulink Design Verifier - The change in enabled status of Proof Objective is not considered while rerunning property proving analysis	09 Jul 2020
	Simulink Design Verifier - The Simulink Design Verifier generated harness model does not simulate with design model having Out Bus Elements of	11 Jun

2172228	heterogeneous types	2020
2167393	Simulink Design Verifier - Incorrect error message with sldvlogsignals when its first argument refers to a model with bus element port blocks	11 Jun 2020
2150560	Simulink Design Verifier - Compatibility check may fail for models containing Subsystem Reference blocks	20 May 2020
2026246	Simulink Design Verifier - Compatibility check may fail for models containing Data Store Memory blocks	20 May 2020
2209498	Simulink Design Verifier - Incorrect objective status reported when a model is analyzed in Accelerator simulation mode	18 May 2020
2202755	Simulink Design Verifier - Simulation mode set to Normal mode, once the analysis is finished	18 May 2020
2202754	Simulink Design Verifier - Fast Restart mode gets disabled after Simulink Design Verifier analysis	18 May 2020
2126877	Simulink Design Verifier - Reusing Simulink cache file errors out when no replacement model is generated for a custom block replacement rule	22 Apr 2020
2179943	Simulink Check - Error message with Function-Call Subsystem added as slice component using addSliceComponent	17 Mar 2020
2118180	Simulink Design Verifier - Compatibility check fails for models with erroneous block specific copy action callbacks	02 Mar 2020

Λ Less

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Simulink PLC Coder*

Display bug reports for Simulink PLC Coder (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 12 Simulink PLC Coder bug reports for release R2020a

ID	Bug Report Summary	Modified
2220730	Simulink PLC Coder - Incorrect Code Generation: Incorrect code generation for PC Worx 6.0 target when using models with Shift Arithmetic blocks	10 Jul 2020
2216089	Simulink PLC Coder - MATLAB might crash when generating PLC code for a model that uses Simulink.Signal	11 Jun 2020
2221963	Simulink PLC Coder - Multi-testbench signal group time range check may cause multi-testbench code generation workflow to error	18 May 2020
2147686	Simulink PLC Coder - Incorrect Code Generation: Incorrect code generated when using y=f(y) style MATLAB function in a Simulink function inside a Stateflow chart	14 May 2020
2201973	Simulink PLC Coder - Incorrect Code Generation: Generated PLC code might produce incorrect results due to automated type conversion from unsigned to signed integer	14 May 2020
2208060	Simulink PLC Coder - Incorrect Code Generation: Code generated for the TIA Portal: Double Precision target IDE could experience inconsistent behavior when type casting a floating-point data type to an integer data type.	22 Apr 2020
2180371	Simulink PLC Coder - Simulink PLC Coder does not support the Simulink.LookupTable, Simulink.Breakpoint, and Simulink.DualScaledParameter objects for code generation	22 Apr 2020
	Simulink PLC Coder - Incorrect Code Generation: Output variables not updated	22 Apr

2176576	for sub-function block calls related to initialization	2020
2147418	Simulink PLC Coder - Incorrect Code Generation: Incorrect code generation for CODESYS target when using models with Shift Arithmetic blocks	22 Apr 2020
2092179	Simulink PLC Coder - Code generation errors out for tunable parameters having fixed-point data type	22 Apr 2020
2182040	Simulink PLC Coder - Simulink PLC coder throws a typecast assertion during code generation	22 Apr 2020
2062037	Simulink PLC Coder - Incorrect Code Generation: PLC Coder generates wrong code for the Discrete-time Integrator block using unsupported integrator methods	11 Oct 2019

∧ Less

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Simulink Check*

Display bug reports for Simulink Check (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 10 Simulink Check bug reports for release R2020a

ID	Bug Report Summary	Modified
2282734	Simulink Check - MAAB check Stateflow transition appearance incorrectly flags transitions crossing junctions	13 Jul 2020
2255599	Simulink Check - The Model Advisor check Check for optimal bus virtuality (ID: mathworks.design.OptBusVirtuality) flags virtual bus crossing model boundary	17 Jun 2020
2227557	Simulink Check - Model Advisor check Check usage of Merge block flags Initialize Function block	20 May 2020
2198087	Simulink Coverage - Incorrect execution coverage for referenced export-function model	18 May 2020
2181624	Simulink Check - Model Transformer tool generates an error while refactoring a model to eliminate Data Store Memory blocks	15 May 2020
2231694	Simulink Check - Model Advisor check Data type selection for index signals produces an error	13 May 2020
2173909	Simulink Requirements - Bullet points not imported correctly from DOORS 9	22 Apr 2020
2195350	Simulink Check - The check for JMAAB Check Stateflow transition appearance (mathworks.jmaab.db_0129) displays an incorrect warning	26 Mar 2020
2179943	Simulink Check - Error message with Function-Call Subsystem added as slice component using addSliceComponent	17 Mar 2020
2172579	Simulink Check - Incorrect warning with Check use of default variants(mathworks.maab.na_0036) for Label variant control mode	24 Feb 2020

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value

Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Simulink Coverage*

Display bug reports for Simulink Coverage (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 6 Simulink Coverage bug reports for release R2020a

ID	Bug Report Summary	Modified
2247819	Simulink Coverage - An error occurs in the Simulink Test Manager while aggregating coverage data for a Subsystem Harness if the subsystem contains a call to an external MATLAB file	10 Jul 2020
2267735	Simulink Coverage - Scoping coverage to requirements-based tests causes 0% coverage for subsystem test harnesses	10 Jul 2020
2179804	Simulink Coverage - An error occurs when a Simulink Subsystem Harness contains a block and subsystem with identical names	05 Jun 2020
2198087	Simulink Coverage - Incorrect execution coverage for referenced export-function model	18 May 2020
2183237	Simulink Coverage - Generating a model coverage report using an exported object can cause an error	26 Mar 2020
2179943	Simulink Check - Error message with Function-Call Subsystem added as slice component using addSliceComponent	17 Mar 2020

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Simulink Test*

Display bug reports for Simulink Test (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 20 Simulink Test bug reports for release R2020a

ID	Bug Report Summary	Modified
2224093	Simulink Test - Cannot override logging for data store defined in data dictionary using Test Manager.	10 Jul 2020
2252259	Simulink Test - Iterations configured for Fast Restart mode run in Normal mode	10 Jul 2020
2249535	Simulink Test - Test result report that includes Signal Editor block data values produces an error	10 Jul 2020
2241749	Simulink Test - Running R2015a test cases with mapped inputs in R2020a might fail in Test Manager	10 Jul 2020
2255433	Simulink Test - Loading externally saved test harness using load_system might cause MATLAB to crash.	01 Jul 2020

2236833	Simulink Test - Recovered Stateflow Charts block inserted in Subsystem Reference test harness	18 Jun 2020
2237793	Simulink Test - Changed ports in an observer model do not highlight correctly in Manager Observer dialog box	11 Jun 2020
2249557	Simulink Test - Running a test file containing test cases with external test harnesses that contain a Signal Builder block might error	11 Jun 2020
2236006	Simulink Test - Test using sltest.testmanager.run on models with fast restart fail, but pass when using Test Manager	11 Jun 2020
2248003	Simulink Test - Testing a component in a library when simulation mode is overridden to not use model settings might fail	11 Jun 2020
2239108	Simulink Test - Test execution compiles model multiple times	11 Jun 2020
2237774	Simulink Test - Dragging ports of a subsystem interface might cause lost connections in associated test harnesses	11 Jun 2020
2204045	Simulink Test - MATLAB might crash when capturing a baseline to a spreadsheet	11 Jun 2020
2212150	Simulink Test - Incorrect override of parameters in Simulink Test	29 May 2020
2160783	Simulink Test - Observer port moved to new signal shows link to original signal	14 May 2020
2210475	Simulink Test - Test suite and test file cleanup callbacks are executed before all test cases are complete	14 May 2020
2194996	Simulink Test - Results export or import fails when custom criteria diagnostic contains a null character	01 Apr 2020
2120213	Simulink Test - Comparison results for complex signals produce "Signals not aligned" warning	13 Dec 2019
2114999	Simulink Test - Running test harnesses using Run with Stepper button on toolstrip is not supported	13 Dec 2019
2112483	Simulink Test - Test that overrides Signal Editor scenario and includes inputs in the results produces an error	13 Dec 2019

Λ Less

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for Simulink Requirements*

Display bug reports for Simulink Requirements (R2020a) available at https://www.mathworks.com/support/bugreports .

NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

There are 22 Simulink Requirements bug reports for release R2020a

ID	Bug Report Summary	Modified
2253967	Simulink Requirements - MATLAB stops responding after updating previously imported requirements in Requirements Editor	10 Jul 2020
2251452	Simulink Requirements - ReqIF ID values might change between revisions when exporting to ReqIF	10 Jul 2020
2222794	Simulink Requirements - Traceability link from Requirement to Simulink Test Case appears unresolved	10 Jul 2020
2172030	Simulink Requirements - Requirements Editor becomes slow when opening requirement sets with large number of incoming links	12 Jun 2020
	Simulink Requirements - 3rd-Party requirements tool does not accept ReqIF	11 Jun

2210749	exported by Simulink Requirements	2020
2247724	Simulink Requirements - Failure to connect with IBM DOORS Next (DNG) when importing requirements	11 Jun 2020
2247892	Simulink Requirements - Traceability Matrix does not render link icons correctly	11 Jun 2020
2182761	Simulink Requirements - Requirements Editor might become frozen after deleting multiple objects	19 May 2020
2210569	Simulink Requirements - Unrecgonized date-time format error when importing DOORS module	19 May 2020
2198087	Simulink Coverage - Incorrect execution coverage for referenced export-function model	18 May 2020
2232550	Simulink Requirements - Displayed column widths in Requirements Editor might be reset	14 May 2020
2206550	Simulink Requirements - Requirements report displays unrelated text with requirement description	22 Apr 2020
2205640	Simulink Requirements - Matlab crashes while updating requirement from IBM DOORS Next server	22 Apr 2020
2200430	Simulink Requirements - Import from IBM DOORS Next Generation broken for non-default server instance	22 Apr 2020
2192264	Simulink Requirements - Simulink Requirements exported ReqIF file has wrong attribute definition references	22 Apr 2020
2191769	Simulink Requirements - Requirements links lost in round trip workflow when exporting with ReqIF	22 Apr 2020
2173909	Simulink Requirements - Bullet points not imported correctly from DOORS 9	22 Apr 2020
2172917	Simulink Requirements - Links to imported DOORS Next items not listed in browser popup window on DOORS Next side	22 Apr 2020
2161457	Simulink Requirements - Import or Update from Microsoft Word fails with an error popup	22 Apr 2020
2179943	Simulink Check - Error message with Function-Call Subsystem added as slice component using addSliceComponent	17 Mar 2020
2163041	Simulink Requirements - Missing requirement links for Stateflow objects in library after resolve-push	06 Feb 2020
	Simulink Requirements - Error when clicking Show in document for references	24 Apr

1970160	imported from IBM Rational DOORS Next Generation module	2019

∧ Less

Recommended Action

To determine if your model is affected by the bugs, review the bug report descriptions and workarounds in the links provided in the ID column.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for AUTOSAR Blockset*

Display bug reports for AUTOSAR Blockset (R2020a) available at https://www.mathworks.com/support/bugreports .
NOTE: This check does not determine whether your model might be affected by these bugs.

Warning
Unable to query bug reports from Mathworks.

Recommended Action
Make sure your internet connection works and you are able to access https://www.mathworks.com/support/bugreports/.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

⚠ *Display bug reports for HDL Coder*

Display bug reports for HDL Coder (R2020a) available at https://www.mathworks.com/support/bugreports .
NOTE: This check does not determine whether your model might be affected by these bugs.

Warning

Unable to query bug reports from Mathworks.

Recommended Action

Make sure your internet connection works and you are able to access https://www.mathworks.com/support/bugreports/.

Input Parameters Selection

Name	Value
Only show bug reports modified after date(mm/dd/yyyy)	

Modeling Standards for EN 50128 ✓0 ✗0 ⚠0 ▤106

Display configuration management data

Not Run

Display model metrics and complexity report

Not Run

Check for unconnected objects

Not Run

High-Integrity Systems ✓0 ✗0 ⚠0 ▤89

Simulink ✓*0* ✗*0* ⚠*0* ▤*28*

Check usage of Abs blocks

Not Run

Check usage of Math Function blocks (rem and reciprocal functions)

Not Run

Check usage of Math Function blocks (log and log10 functions)

Not Run

Check usage of While Iterator blocks

Not Run

Check usage of For and While Iterator subsystems

Not Run

Check usage of For Iterator blocks

Not Run

Check usage of If blocks and If Action Subsystem blocks

Not Run

Check usage of Switch Case blocks and Switch Case Action Subsystem blocks

Not Run

Check usage of conditionally executed subsystems

Not Run

Check usage of Merge blocks

Not Run

Check Relational Operator blocks equating floating-point types

Not Run

Check usage of Relational Operator blocks

Not Run

Check usage of Logical Operator blocks

Not Run

Check usage of bit operation blocks

Not Run

Check for blocks not recommended for C/C++ production code deployment

Not Run

Check for inconsistent vector indexing methods

Not Run

Check data types for blocks with index signals

Not Run

Check usage of variant blocks

Not Run

Check usage of lookup table blocks

Not Run

Check usage of Signal Routing blocks

Not Run

Check for root Inports with missing properties

Not Run

Check for root Inports with missing range definitions

Not Run

Check for root Outports with missing range definitions

Not Run

Check usage of Assignment blocks

Not Run

Check global variables in graphical functions

Not Run

Check usage of Gain blocks

Not Run

Check for length of user-defined object names

Not Run

Check data type of loop control variables

Not Run

Stateflow ✓*0* ✗*0* ⚠*0* ☰*14*

Check state machine type of Stateflow charts

Not Run

Check Stateflow charts for ordering of states and transitions

Not Run

Check usage of bitwise operations in Stateflow charts

Not Run

Check for Strong Data Typing with Simulink I/O

Not Run

Check Stateflow debugging options

Not Run

Check Stateflow charts for transition paths that cross parallel state boundaries

Not Run

Check for inappropriate use of transition paths

Not Run

Check Stateflow charts for strong data typing

Not Run

Check naming of ports in Stateflow charts

Not Run

Check scoping of Stateflow data objects

Not Run

Check Stateflow charts for uniquely defined data objects

Not Run

Check usage of shift operations for Stateflow data

Not Run

Check assignment operations in Stateflow charts

Not Run

Check Stateflow charts for unary operators

Not Run

MATLAB ✓*0* ✗*0* ⚠*0* ☰*10*

Check usage of standardized MATLAB function headers

Not Run

Check for MATLAB Function interfaces with inherited properties

Not Run

Check MATLAB Function metrics

Not Run

Check MATLAB Code Analyzer messages

Not Run

Check if/elseif/else patterns in MATLAB Function blocks

Not Run

Check switch statements in MATLAB Function blocks

Not Run

Check usage of relational operators in MATLAB Function blocks

Not Run

Check usage of equality operators in MATLAB Function blocks

Not Run

Check usage of logical operators and functions in MATLAB Function blocks

Not Run

Check type and size of condition expressions

Not Run

Configuration ✓*0* ✗*0* ⚠*0* ▤*32*

Check safety-related diagnostic settings for data store memory

Not Run

Check safety-related diagnostic settings for saving

Not Run

Check safety-related model referencing settings

Not Run

Check safety-related code generation settings for comments

Not Run

Check safety-related code generation interface settings

Not Run

Check safety-related solver settings for simulation time

Not Run

Check safety-related solver settings for solver options

Not Run

Check safety-related solver settings for tasking and sample-time

Not Run

Check safety-related diagnostic settings for solvers

Not Run

Check safety-related diagnostic settings for sample time

Not Run

Check safety-related optimization settings for logic signals

Not Run

Check safety-related block reduction optimization settings

Not Run

Check safety-related code generation settings for code style

Not Run

Check safety-related optimization settings for application lifespan

Not Run

Check safety-related code generation identifier settings

Not Run

Check safety-related optimization settings for loop unrolling threshold

Not Run

Check safety-related optimization settings for data initialization

Not Run

Check safety-related optimization settings for data type conversions

Not Run

Check safety-related optimization settings for division arithmetic exceptions

Not Run

Check safety-related optimization settings for specified minimum and maximum values

Not Run

Check safety-related diagnostic settings for compatibility

Not Run

Check safety-related diagnostic settings for parameters

Not Run

Check safety-related diagnostic settings for Merge blocks

Not Run

Check safety-related diagnostic settings for model initialization

Not Run

Check safety-related diagnostic settings for data used for debugging

Not Run

Check safety-related diagnostic settings for signal connectivity

Not Run

Check safety-related diagnostic settings for bus connectivity

Not Run

Check safety-related diagnostic settings that apply to function-call connectivity

Not Run

Check safety-related diagnostic settings for type conversions

Not Run

Check safety-related diagnostic settings for model referencing

Not Run

Check safety-related diagnostic settings for Stateflow

Not Run

Check safety-related diagnostic settings for signal data

Not Run

Naming ✓*0* ✗*0* ⚠*0* ☰*2*

Check model file name

Not Run

Check model object names

Not Run

Requirements ✓*0* ✗*0* ⚠*0* ☰*1*

Check for model elements that do not link to requirements

Not Run

Code ✓*0* ✗*0* ⚠*0* ☰*2*

Check for blocks not recommended for MISRA C:2012

Not Run

Check configuration parameters for MISRA C:2012

Not Run

Bug Reports ✓0 ✗0 ⚠0 ☰14

Display bug reports for Embedded Coder

Not Run

Display bug reports for IEC Certification Kit

Not Run

Display bug reports for Polyspace Code Prover

Not Run

Display bug reports for Polyspace Bug Finder

Not Run

Display bug reports for Polyspace Code Prover Server

Not Run

Display bug reports for Polyspace Bug Finder Server

Not Run

Display bug reports for Simulink Design Verifier

Not Run

Display bug reports for Simulink PLC Coder

Not Run

Display bug reports for Simulink Check

Not Run

Display bug reports for Simulink Coverage

Not Run

Display bug reports for Simulink Test

Not Run

Display bug reports for Simulink Requirements

Not Run

Display bug reports for AUTOSAR Blockset

Not Run

Display bug reports for HDL Coder

Not Run

Model Metrics ✓0 ✗0 ⚠0 ▤10

📁 Count Metrics ✓0 ✗0 ⚠0 ☰7

☰ *Simulink block metric*

Not Run

☰ *Subsystem metric*

Not Run

☰ *Library link metric*

Not Run

☰ *Effective lines of MATLAB code metric*

Not Run

☰ *Stateflow chart objects metric*

Not Run

☰ *Lines of code for Stateflow blocks metric*

Not Run

☰ *Subsystem depth metric*

Not Run

📁 Complexity Metrics ✓0 ✗0 ⚠0 ☰1

☰ *Cyclomatic complexity metric*

Not Run

📁 Readability Metrics ✓0 ✗0 ⚠0 ☰2

☰ *Nondescriptive block name metric*

Not Run

☰ *Data and structure layer separation metric*

Not Run

🗀 Modeling Signals and Parameters using Buses ✓0 ✗0 ⚠0 ☰3

☰ Check for optimal bus virtuality

Not Run

☰ Check structure parameter usage with bus signals

Not Run

☰ Check bus signals treated as vectors

Not Run

🗀 Code Generation Efficiency ✓9 ✗0 ⚠2 ☰0

⚠ Check optimization settings

Check optimization settings

Warning

The model configuration parameters are not set to the recommended values specified in the data file.

Status	Parameter	Current Value	Recommended Values
Warning	Block reduction (BlockReduction)	off	on
Warning	Inline invariant signals (InlineInvariantSignals)	off	on
Warning	Simulation range checking (SignalRangeChecking)	error	none

Warning	Ignore test point signals (IgnoreTestpoints)	off	on

Recommended Action

Follow the links in the result table to modify the model configuration parameters.

✓ Identify blocks using one-based indexing

Check the model for blocks configured for one-based indexing

Passed

All blocks in the model use zero-based indexing.

✓ Identify questionable software environment specifications

Passed

✓ Identify lookup table blocks that generate expensive out-of-range checking code

These settings can lead to inefficient code generation when inputs always fall within the range of valid breakpoint values for lookup table blocks, including Lookup Table blocks, Prelookup blocks and Interpolation blocks.

Passed

The lookup table blocks have been configured to generate range-checking free code.

✓ Identify questionable code instrumentation (data I/O)

Passed

✓ Check output types of logic blocks

Identify logic blocks that are outputting non-Boolean data types.

Passed

All logic blocks are being used appropriately.

✓ Check configuration parameters for generation of inefficient saturation code

Check setting for generation of saturation code which handles out of range values. This code decreases the net efficiency of the application.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Remove code from tunable parameter expressions that saturates out-of-range values (EfficientTunableParamExpr)	on	on	IsERTTarget
D - Pass	Target derived from ERT (IsERTTarget)	on	on	
Pass	Remove code from floating-point to integer conversions with saturation that maps NaN to zero (EfficientMapNaN2IntZero)	on	on	IsERTTarget

---

✓ Identify blocks that generate expensive rounding code

Check for expensive rounding operations in multiplication and division

Passed

Check Optimization and Hardware Implementation settings (Lookup Blocks)

Passed

Check for expensive rounding in a data type conversion

Passed

Check for expensive rounding modes in the model

Passed

✓ Identify questionable fixed-point operations

Check for multiword operations

Passed

Check for expensive multiplication code

Passed

Check for expensive division code

Passed

Identify lookup blocks with uneven breakpoint spacing

Passed

Check for expensive pre-lookup division

Passed

Check for expensive data type conversions

Passed

Check for fixed-point comparisons with predetermined results

Passed

Check for expensive binary comparison operations

Passed

Check for expensive fixed-point types

Passed

⚠ Identify blocks that generate expensive fixed-point and saturation code

Identify Sum blocks for questionable fixed-point operations

Passed

Identify Relational Operation blocks for questionable fixed-point operations

Passed

Identify Data Type Conversion Inherited blocks for questionable fixed-point operations

Passed

Identify Switch blocks for questionable fixed-point operations

Passed

Identify Logic blocks for questionable fixed-point operations

Passed

Identify Saturate blocks for questionable fixed-point operations

Passed

Identify Min Max blocks for questionable fixed-point operations

Passed

Identify Discrete Integrator blocks for questionable fixed-point operations

Passed

Identify Compare To Constant blocks for questionable fixed-point operations

Passed

Identify Lookup Table blocks for questionable fixed-point operations

Passed

Identify blocks that will invoke net slope computation

Passed

Identify Product blocks that are less efficient

Passed

Check for expensive saturation code

Warning

Check Optimization and Hardware Implementation settings

Integer division generated code will contain protection against arithmetic exceptions like division by zero, INT_MIN/-1, and LONG_MIN/-1. Depending on the hardware, arithmetic exceptions can cause dramatic behavior, so protecting against these exceptions can be critical. Some designers construct their models in an extremely careful way that makes it impossible for exception triggering input combinations to reach a division operation. For these models, the protection code generated as part of the division operation is redundant. This redundant protection code can be removed using the fixed-point division setting in the Configuration Parameters dialog box. This should only be done when you have painstakingly verified that your model cannot cause exceptions in division operations.

Remove code that protects against division arithmetic exceptions

- Wrong_Way_Driver_Warning

✓ Identify blocks generating inefficient algorithms

Passed

No inefficient algorithms found in the model.

Modeling Single-Precision Systems ✓0 ✗0 ⚠0 ☰1

☰ Identify questionable operations for strict single-precision design

Not Run

Migrating to Simplified Initialization mode ✓0 ✗0 ⚠0 ☰4

☰ Check usage of Merge blocks

Not Run

☰ Check usage of Outport blocks

Not Run

☰ Check usage of Discrete-Time Integrator blocks

Not Run

Check model settings for migration to simplified initialization mode

Not Run

Row-major code generation ✓0 ✗0 ⚠0 ☰3

Identify blocks generating inefficient algorithms

Not Run

Check for blocks not supported for row-major code generation

Not Run

Identify TLC S-Functions with unset array layout

Not Run

Model Referencing ✓0 ✗0 ⚠0 ☰8

Check for model reference configuration mismatch

Not Run

Check diagnostic settings ignored during accelerated model reference simulation

Not Run

Check code generation identifier formats used for model reference

Not Run

Check for parameter tunability information ignored for referenced models

Not Run

Check for implicit signal resolution

Not Run

▤ Check bus signals treated as vectors

Not Run

▤ Check root model Inport block specifications

Not Run

▤ Check for large number of function arguments from virtual bus across model reference boundary

Not Run

Managing Library Links And Variants ✓0 ✗0 ⚠0 ▤4

▤ Identify disabled library links

Not Run

▤ Identify parameterized library links

Not Run

▤ Identify unresolved library links

Not Run

▤ Identify configurable subsystem template blocks having the instances in the model for converting to variant subsystem blocks.

Not Run

Data Transfer Efficiency ✓0 ✗0 ⚠0 ▤1

▤ Check Delay, Unit Delay and Zero-Order Hold blocks for rate transition

Not Run

S-function Checks ✓0 ✗0 ⚠0 ▤1

Check S-functions in the model

Not Run

Simulink Design Verifier Compatibility Check ✓0 ✗0 ⚠0 ☰1

Check compatibility with Simulink Design Verifier

Not Run

Simulink Design Verifier Design Error Checks ✓0 ✗0 ⚠0 ☰9

Detect Dead Logic

Not Run

Detect Out Of Bound Array Access

Not Run

Detect Division By Zero

Not Run

Detect Integer Overflow

Not Run

Detect Non-finite and NaN Floating-point Values

Not Run

Detect Subnormal Floating-point Values

Not Run

Detect Specified Minimum and Maximum Value Violations

Not Run

▤ Detect Data Store Access Violations

Not Run

▤ Detect Block Input Range Violations

Not Run

Simulation Accuracy ✓0 ✗0 ⚠0 ▤1

▤ Check for non-continuous signals driving derivative ports

Not Run

Simulation Runtime Accuracy Diagnostics ✓0 ✗0 ⚠0 ▤2

▤ Runtime diagnostics for S-functions

Not Run

▤ Check if Read/Write diagnostics are enabled for Data Store blocks

Not Run

Managing Data Store Memory Blocks ✓3 ✗0 ⚠0 ▤0

✓ Check Data Store Memory blocks for multitasking, strong typing, and shadowing issues

Passed

✓ Check data store block sample times for modeling errors

Passed

✓ Check for potential ordering issues involving data store access

Passed

Simulink Model File Integrity ✓0 ✗0 ⚠0 ▤2

▤ Check model for foreign characters

Not Run

▤ Check Model History properties

Not Run

Requirements Consistency Checking ✓0 ✗0 ⚠0 ▤4

▤ Identify requirement links with missing documents

Not Run

▤ Identify requirement links that specify invalid locations within documents

Not Run

▤ Identify selection-based links having description fields that do not match their requirements document text

Not Run

▤ Identify requirement links with path type inconsistent with preferences

Not Run

Modeling Standards for MAB ✓126 ✗0 ⚠15 ▤0

Naming Conventions ✓19 ✗0 ⚠1 ▤0

General ✓*4* ✗*0* ⚠*0* ▤*0*

✓ Check file names

Check the model file name to ensure that the name complies with the recommended guidelines.

Passed

All files have correct names.

✓ Check folder names

Check the folder name to ensure that the name complies with the recommended guidelines.

Passed

All folders have correct names.

✓ Check length of model file name

Check length of model file name

Passed

Model name is valid.

✓ Check length of folder name at every level of model path

The model file name is: Wrong_Way_Driver_Warning

Passed

Folder names are valid.

🗀 *Content* ✓*15* ✗*0* ⚠*1* ☰*0*

✓ Check subsystem names

Identify subsystem names with incorrect characters.

Passed

All the subsystem names use correct characters.

✓ Check port block names

Identify Inport or Outport block names with incorrect characters.

Passed

All the Inport or Outport block names use correct characters.

✓ Check character usage in block names

Identify block names with incorrect characters.

Passed

All the block names use correct characters.

✓ Check length of subsystem names

Check length of subsystem names

Passed

All subsystem names are valid.

✓ Check length of block names

Check length of block names

Passed

All block names are valid.

✓ Check length of Inport and Outport names

Check length of Inport and Outport names

Passed

All Inport and Outport names are valid.

✓ Check usable characters for signal names and bus names

Identify invalid characters in signal and bus names

Passed

No invalid characters are used in signal and bus names.

✓ Check usable characters for parameter names

Identify invalid characters in parameter names

Passed

No invalid characters are used in parameter names.

✓ Check length of signal and bus names

Check length of signal and bus names

Passed

All signal and bus names are valid.

✓ Check length of parameter names

Check length of parameter names

Passed

All parameter names are valid.

✓ Check usable characters for Stateflow data names

Identify invalid characters in Stateflow data names.

Passed

No invalid characters are used in Stateflow data names.

✓ Check length of Stateflow data name

Check if the length of Stateflow data names are within limit.

Passed

All Stateflow data names are valid.

✓ Check duplication of Simulink Data names

Simulink Data names should be unique across base workspace, model workspace and data dictionary.

Passed

All Simulink Data names are unique.

⚠ Check unused data in Simulink Model

Check for unused data in Data Dictionary

Warning

The following data variables in the data dictionary are unused:

Data Objects	**Source**
F32	WWDW_dd.sldd
F64	WWDW_dd.sldd
FLAG	WWDW_dd.sldd
MODE	WWDW_dd.sldd
S16	WWDW_dd.sldd
S32	WWDW_dd.sldd
S8	WWDW_dd.sldd
SMODE	WWDW_dd.sldd
SignPosLat	WWDW_dd.sldd
Sign_Status	WWDW_dd.sldd
U16	WWDW_dd.sldd
U32	WWDW_dd.sldd
U8	WWDW_dd.sldd
ValidLateralPosition	WWDW_dd.sldd
ValidLongitudionalPosition	WWDW_dd.sldd
VehicleSpeed_kph	WWDW_dd.sldd
VehicleYawRate_Degree	WWDW_dd.sldd

∧ Less

Recommended Action

Consider removing the unused data variables.

✓ Check for unused data in Stateflow Charts

Checks if the model parameter 'Unused data, events, messages and functions' is not set to 'none'.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unused data, events, messages and functions (SFUnusedDataAndEventsDiag)	warning	error, warning

---

✓ Check usage of restricted variable names

Checks whether variable names used in MATLAB Function blocks are reserved for C/C++/MATLAB keywords

Passed

No variable names conflict with reserved keywords

Simulink ✓48 ✗0 ⚠14 ▤0

Configuration Parameters ✓*4* ✗*0* ⚠*0* ▤*0*

✓ Check Implement logic signals as Boolean data (vs. double)

Identify whether **Implement logic signals as Boolean data (vs. double)** is selected.

Passed

Implement logic signals as Boolean data (vs. double) is selected.

✓ Check Signed Integer Division Rounding mode

jc_0642: Integer rounding mode setting

Identifies blocks with block parameter 'Integer Rounding Mode' set to 'Simplest' when the configuration parameter 'Signed integer division rounds to' is set to 'Undefined'.

Passed

Configuration parameter 'Signed integer division rounds to' is not set to 'Undefined'.

✓ Check diagnostic settings for incorrect calculation results

Identify data validity diagnostic settings which detect incorrect calculation results.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Division by singular matrix (CheckMatrixSingularityMsg)	error	error
Pass	Inf or NaN block output (SignalInfNanChecking)	error	error
Pass	Wrap on overflow (IntegerOverflowMsg)	error	error
Pass	Saturate on overflow (IntegerSaturationMsg)	error	error

✓ Check model diagnostic parameters

Identify diagnostic parameters that are set to `none`.

Passed

All of the diagnostic parameters are set to `error` or `warning`.

Diagram Appearance ✓*13* ✗*0* ⚠*4* ▤*0*

⚠ Check for Simulink diagrams using nonstandard display attributes

Identify nonstandard display attributes in Simulink diagrams.

Check format settings

Identify incorrect model-level format options.

Warning

The following format display options are incorrect.

Display Attribute	Recommended Value	Actual Value
Debug > Information Overlays > Nonscalar Signals	on	off
Debug > Information Overlays > Port Data Type	off	on
Modeling > Environment > Model Browser	off	on
Debug > Information Overlays > Colors	none	disabled

Recommended Action

Set the format options to the recommended value.

Check block colors

Identify blocks using nonstandard colors.

Passed

All blocks use standard colors.

Check canvas colors

Identify canvases that are not white.

Passed

All diagrams use a white canvas.

Check diagram zoom

Identify diagrams that do not have zoom factor set to 100 %.

Note: Zoom factors can differ for each instance of a model diagram opened in Simulink Editor

Warning

The following diagrams do not have zoom factor set to 100 percent:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner

Recommended Action

For each listed diagram, select **Modeling > Environment > Zoom > Normal View (100%)**.

✓ Check Model font settings

Identify blocks and charts with different font settings from input parameters.

Passed

Font settings of the blocks and charts and input parameters are same.

✓ Check whether block names appear below blocks

Identify blocks where the name is not displayed below the block.

Passed

All blocks have names displayed below the block.

✓ Check the display attributes of block names

Identify whether to display block names.

Check for blocks with hidden names and obvious function

Identify block names that are displayed but can be hidden due to obvious behavior.

Passed

All blocks with obvious behavior have hidden names.

Check for non-descriptive displayed block names

Identify block names that are displayed but should be hidden due to a lack of a descriptive name.

Passed

All displayed names provide descriptive information.

Check for missing block names

Identify block names that are hidden but should be displayed to show a descriptive name.

Passed

All displayed names provide descriptive information.

✓ Check for nondefault block attributes

Identify blocks that use and fail to display nondefault values.

Passed

Model displays all block parameter values that are not default values.

⚠ Check Model Description

Identify layers in the model having inconsistent description format.

Warning

Following layers do not have model descriptions:

- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData
-
-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
-
-
-
-
-

∧ Less

Recommended Action

Consider adding model description for all the layers.

✓ Check if blocks are shaded in the model

Check if blocks are shaded in the model

Passed

Blocks in the model are not shaded.

✓ Check for unconnected ports and signal lines

Identify unconnected block input ports, output ports, and signal lines.

Passed

All lines and ports in the model are connected.

⚠ Check signal line connections

Check signal intersections

Warning

The following signals intersect with other signals in the diagram:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change

Recommended Action

Reposition the above listed signals to avoid intersections.

✓ Check signal flow in model

Identify subsystems which do not have a signal flow from left to right.

Passed

No subsystems found with inappropriate signal flow.

✓ Check usage of tunable parameters in blocks

Identify tunable parameters used to specify expressions, data type conversions, or indexing operations.

Passed

Tunable parameters are not used in the model.

✓ Check connections between structural subsystems

Identify connections between structural subsystems.

Passed

All connections to structural subsystems adhere to the guideline.

✓ Check for consistency in model element names

Check if model elements connected to a signal are following consistent naming.

Passed

Model elements connected to a signal are following consistent names.

✓ Check trigger signal names

Identify trigger blocks where the origin of the trigger signal and the destination have dissimilar names.

Passed

No violation of the guideline for use of trigger signal names.

⚠ Check for mixing basic blocks and subsystems

Identify levels in the model that include basic blocks and subsystems. Each level of a model must be designed with blocks of the same level (for example, only subsystems or only basic blocks).

Warning

The following level(s) in the model include basic blocks and subsystems:

System	Block path
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Sign	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant" title="Wrong_Way_Driver_Warning/Input_Signal_Ran

al_Range_Conditioner/Valid_Signal	ge_Conditioner/Valid_Signal/Constant
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Ran

	ge_Conditioner/Valid_Signal/Relational Operator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator3

Recommended Action

If possible, replace blocks at the identified level of the model hierarchy with basic blocks. Move nonvirtual blocks into the identified subsystem.

✓ Check for avoiding algebraic loops between subsystems
jc_0653: Guidelines for avoiding algebraic loops between subsystems.

Passed
No delay blocks in feedback loops violate the guidelines for avoiding algebraic loops between subsystems.

✓ Check for prohibited sink blocks
Identify sink blocks that must be removed prior to code generation.

Passed
There are no prohibited blocks in the subsystem.

📁 *Signal* ✓*3* ✗*0* ⚠*3* ☰*0*

✓ Check usage of vector and bus signals

Identify mixed usages of vector and bus signals.

Passed

No mixing of vector and bus signals found in the system.

⚠ Check signal line labels

Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check source block labels

Warning

The following signals have no label:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant2/

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant3/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant4/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant5/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant6/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3/

Λ Less

Recommended Action

Add a new or propagated label to the signal line.

____ Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check destination block labels

Warning

The following signals have no label:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/

Recommended Action

Add a new or propagated label to the signal line.

✓ Check for propagated signal labels

Identify propagated labels on signal lines.

Passed

All inputs and outputs to the subsystems and blocks have labels and display propagated signals.

⚠ Check position of signal labels

Check location of signal labels

Warning

The following signals do not have labels located at the origin of the signal line:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning

- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning

∧ Less

Recommended Action

Consider placing the labels at the origin of the signal line.

⚠ Check signal line labels

Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check source block labels

Warning

The following signals have no label:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant2/

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant3/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant4/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant5/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant6/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3/

Λ Less

Recommended Action

Add a new or propagated label to the signal line.

____ Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check destination block labels

Warning

The following signals have no label:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/

Recommended Action

Add a new or propagated label to the signal line.

✓ Check for propagated signal labels

Identify propagated labels on signal lines.

Passed

All inputs and outputs to the subsystems and blocks have labels and display propagated signals.

🗀 *Block Consistency* ✓*4* ✗*0* ⚠*2* 🗎*0*

✓ Check Indexing Mode

Identify blocks and charts with inconsistent Indexing mode.

Passed

No inconsistent Indexing mode used in the model.

✓ Check block orientation

Identify blocks which are rotated or reversed

Passed

No blocks found with rotated or reversed orientation

⚠ Check if tunable block parameters are defined as named constants

Check if tunable block parameters are defined as named constants

Warning

The following tunable block parameters are not defined as named constants.

Block	Violations
	Value : -60
	Value : 60
	Value : 255

Recommended Action

Consider changing tunable block parameter literal values to named constants.

✓ Check for sample time setting

Check if sample time property of a block is set to -1 (inherited).

Passed

All permitted blocks have sample time set to -1 (inherited).

✓ Check usage of fixed-point data type with non-zero bias

jc_0643: Fixed-point setting

Identify blocks with a fixed-point data type whose bias is not zero.

Passed

No blocks found with the Data Type Assistant mode set to "Fixed point" and a bias value other than zero

---

⚠ Check type setting by data objects

jc_0644: Identify blocks that violate signal data type setting if signal objects are used.

Warning

The following blocks violate signal data type setting if signal objects are used.

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Switch
-
-

Recommended Action

Set the output data type of the blocks either to "auto" or "Inherit via back propagation". This check excludes Data Type Conversion block, type setting by fixdt, double and boolean data types, and reusable internal part of function (treat as atomic unit).

Conditional Subsystem relations ✓*5* ✗*0* ⚠*0* ▤*0*

✓ Check position of conditional blocks and iterator blocks

Identify conditional and iterative blocks that are positioned inconsistently in the model.

Passed

The conditional and iterative blocks are correctly placed in the model.

✓ Check undefined initial output for conditional subsystems

Check undefined initial output for Outports/Merge blocks in conditional subsystems

Passed

The initial output setting for all Conditional Subsystems are valid.

✓ Check usage of Merge block

jc_0659: Usage restrictions of signal lines inputted to Merge block
There must not be any block between a Conditional Subsystem block and a Merge block.

Passed

No Merge block found.

✓ Check logical expressions in If blocks

Checks If blocks for complex usage of primary expressions within a logical expression

Passed

Logical expressions inside If blocks are simple

✓ Check default/else case in Switch Case blocks and If blocks

Check if default/else case in Switch Case blocks and If blocks are set to 'on'

Passed

Conditional Control blocks are valid.

Operation Blocks ✓*11* ✖*0* ⚠*3* ▤*0*

⚠ Check fundamental logical and numerical operations

Check input data types of blocks meant for numerical operations

Warning

The following numerical operation blocks have boolean data type as input:

-
-
-
-
-
-
-
-
-

Recommended Action

Consider having non-boolean inputs for the numerical operation blocks.

✓ Check usage of Sum blocks

Identify inappropriate usages of Sum block.

Passed

No violations of the guideline found with the usage of the Sum block.

✓ Check operator order of Product blocks

Operator order for Product blocks.

Passed

All Product blocks have valid operator order.

✓ Check signs of input signals in product blocks

jc_0611: Input signal sign during product block division

Identify blocks that perform division whose inputs have different sign bit.

Passed

No product block with division of different sign bits found.

✓ Check for parentheses in Fcn block expressions

jc_0622: Guideline for using the Fcn block

Passed

All Fcn blocks use parentheses to mark operator precedence.

✓ Check icon shape of Logical Operator blocks

Icon shape of Logical Operator blocks

Passed

All Logical Operator blocks have consistent icon shape.

⚠ Check usage of Relational Operator blocks

Identify Relational Operator blocks that connect to constants with the first (upper) input value.

Warning

The following Relational Operator blocks connect to a constant value using the first (upper) input value:

-
-
-
-

Recommended Action

Make the constant value the second (lower) input to the Relation Operator block.

✓ Comparing floating point types in Simulink

jc_0800: Comparing floating point types in Simulink
Equivalence comparison should not be used for floating point numbers.

Passed

No Equivalence comparison done on floating point numbers.

✓ Check usage of Lookup Tables

jc_0626: Guideline for using the Lookup Table system block
Checks for the recommended parameter settings in Lookup Tables to prevent unexpected results.

Passed

All the Lookup Tables pass the check.

✓ Check usage of Memory and Unit Delay blocks

Identify Memory blocks not using a continuous sample time

Passed

No Memory blocks found with inappropriate sample time

Identify Unit Delay blocks with non-discrete sample time

Passed

No Unit Delay blocks found with non-discrete sample time

✓ Check for cascaded Unit Delay blocks

Identify cascaded and tapped pattern of Unit Delay blocks.

Passed

No cascaded Unit Delay blocks found that can be changed to Tapped Delay/Delay block.

✓ Check usage of Discrete-Time Integrator block

jc_0627: Identify Discrete-Time Integrator blocks that violate saturation limit settings

Passed

No Discrete-Time Integrator blocks found that violate JMAAB guideline jc_0627

✓ Check usage of the Saturation blocks

jc_0628: Identify the Saturation and Saturation Dynamic blocks that perform type casting.

Passed

No Saturation and/or Saturation Dynamic blocks perform type casting

---

⚠ Check output data type of operation blocks

jc_0651: Guideline for implementing a type conversion.

Warning

Following operation blocks explicitly specify output data type:

-
-
-
-

-
-
-
-
-
-
-
-
-
-
-
-
-
-

∧ Less

Recommended Action

Instead of explicitly specifying output data type on operation blocks, use 'Data Type Conversion' block when changing the data type of the block output signal.

📁 *Other blocks* ✓*8* ✗*0* ⚠*2* ▤*0*

⚠ Check position of Inport and Outport blocks

Check positions of Inport blocks

Warning

The following Inport blocks are not placed to left side of the diagram:

-
-

Recommended Action

Move the Inport blocks identified to the left of all other blocks in the diagram.

It is acceptable to move the Inport block to the right only to prevent signal crossings.

Check positions of Outport blocks

Warning

The following Outport blocks are not placed to right side of the diagram:

-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/WarningDisplayHMI
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SignID
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SignRelevance
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SignAboveRoad

Recommended Action

Move the Outport blocks identified to the right of all other blocks in the diagram.

It is acceptable to move the Outport block to the left only to prevent signal crossings.

✓ Check display for port blocks

Identify Inport and Outport blocks that do not specify `Port number` for the **Icon display** block parameter.

Passed

All port blocks display the port number.

✓ Check scope of From and Goto blocks

Identify incorrect scoping of From and Goto blocks. For signal flows, From and Goto blocks must use local scope. Control flow can use global scope.

Passed

All From and Goto blocks are used correctly.

✓ Check for usage of Data Store Memory blocks

Identify the usage of Data Store Memory blocks.

Passed

Usage of Data Store Memory blocks is correct.

⚠ Check usage of Switch blocks

Identify Switch blocks that do not use Boolean inputs for the switch condition (input 2), and do not use `u2 ~= 0` for the **Criteria for passing first input** block parameter.

Check Switch block parameters

Identify Switch blocks with the parameter **Criteria for passing first input** not set to `u2 ~= 0`.

Warning

The block parameter **Criteria for passing first input** is not set to `u2 ~= 0` for the following blocks:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Switch
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch1
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch2
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Switch" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Switch

Recommended Action

Set the block parameter **Criteria for passing first input** to `u2 ~= 0`. This might require reworking the logic associated with the Switch block.

Check for Boolean switch condition
Identify blocks that do not use Boolean signal switch conditions (input 2).

Passed
The switch condition is a Boolean signal.

✓ Check input and output datatype for Switch blocks
jc_0650: Identify Switch blocks with mismatched input and output data types

Passed
No Switch blocks found with mismatched input and output data types

---

✓ Check settings for data ports in Multiport Switch blocks
Identify Multiport Switch blocks that violate data port settings.

Passed
No Multiport Switch blocks found with inappropriate data port settings.

✓ Check for missing ports in Variant Subsystems

Check for number of inputs/outputs to a Variant Subsystem.

Passed

No Variant Subsystems found having different number of inputs/outputs in the Variant Subsystem choices.

✓ Check use of default variants

na_0036: Default variant

Identify variant subsystems that do not use default variants.

Passed

All variant subsystems in the model use default variants

✓ Check use of single variable variant conditionals

Identify variant subsystems which use multi-variable compound conditions

Passed

No variant subsystems with multiple variable compound conditions found

🗀 Stateflow ✓51 ✖0 ⚠0 ▤0

🗀 *Block/Data/Events* ✓*7* ✖*0* ⚠*0* ▤*0*

✓ Check for Strong Data Typing with Simulink I/O

Check whether labeled input and output signals are strongly typed.

Passed

No Stateflow charts have **Use Strong Data Typing with Simulink I/O** cleared.

✓ Check for names of Stateflow ports and associated signals

Identify mismatches between names of Stateflow ports and the associated signals.

Passed

No Stateflow charts were found.

✓ Check execution timing for default transition path

'Execute (enter) Chart At Initialization' should be set to OFF.

Passed

All Stateflow Charts pass the check.

✓ Check definition of Stateflow data

Identify the Scope value set on Stateflow data defined at machine level.

Passed

All Stateflow data at machine level has been defined as per guideline.

✓ Check usable number for first index

Identify usage of first index of Stateflow data.

Passed

All Stateflow data first index values are uniform.

✓ Check scope of data in parallel states

jc_0722: Guidelines for setting local variables in parallel states

The scope of local variables should be restricted to one parallel state unless it is being used by other parallel states.

Passed

No Stateflow States were found.

✓ Check definition of Stateflow events

Stateflow events should be defined at the smallest possible scope of usage.

Passed

All Stateflow events are defined at their smallest scope.

📁 *Diagram* ✓*14* ✗*0* ⚠*0* ☰*0*

✓ Check for unconnected objects in Stateflow Charts

Identify dangling transitions and unconnected Stateflow States and Junctions in Stateflow Charts.

Passed

No unconnected transitions, states or junctions found in Stateflow Charts.

✓ Check usage of exclusive and default states in state machines

Identify Stateflow charts and substates that incorrectly use or define exclusive and default states.

Check Stateflow charts for exclusive states

Identify Stateflow charts that have singular exclusive (OR) states.

Passed

The Stateflow charts do not have singular exclusive (OR) states.

Check Stateflow charts for undefined default states

Identify Stateflow charts that do not define default states.

Passed

Each Stateflow chart defines a default state.

Check for multiple states assigned as the default state

At the root level in the Stateflow hierarchy only one state should be assigned as the default.

Passed

The root level of the chart has only one default state assigned.

Check for substates with singular OR states

States configured as OR should always be part of a group of states.

Passed

No singular OR states were detected.

Check for substates without default states defined

At every level in the Stateflow hierarchy a default state should be assigned.

Passed

All substates have default states assigned.

Check for substates with multiple default states defined

At every level in the Stateflow hierarchy only one state should be assigned as the default.

Passed

All levels of the chart have only one default state assigned.

✓ Check for parallel Stateflow state used for grouping

jc_0721: Guidelines for using parallel states

Identify parallel Stateflow States used for grouping.

Passed

No Stateflow charts were found.

✓ Check Stateflow transition appearance

Identify Stateflow transitions visually overlapping other Stateflow objects.

Passed

No transition violates the guidelines for Stateflow transition appearance.

✓ Check default transition placement in Stateflow charts

jc_0531: Placement of default transition.

Passed

No Stateflow transitions and states found that violate the guidelines for default transition placement in Stateflow charts.

✓ Check usage of transitions to external states

Identify transitions ending on external child states.

Passed

No direct transitions found from external state to child state.

✓ Check for unexpected backtracking in state transitions

Identify configuration parameter settings which identify unexpected backtracking in state transitions.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unexpected backtracking (SFUnexpectedBacktrackingDiag)	error	error

---

✓ Check starting point of internal transition in Stateflow

jc_0760: In all state charts and flow charts, internal transitions from state boundaries must start from the left edge of the state.

Passed

No Stateflow transitions found that violate the guidelines for starting point of internal transition in Stateflow.

✓ Check usage of internal transitions in Stateflow states

Identify Stateflow states using multiple internal transitions.

Passed

No Stateflow states found with multiple internal transitions

✓ Check prohibited combination of state action and flow chart

jc_0762: State actions within states and flow chart statements should not be used in combination.

Passed

No Stateflow states found that combine state action and flow chart.

✓ Check transition orientations in flow charts

Identify transitions in Stateflow flow charts that are drawn incorrectly.

Check for conditions drawn horizontally

Condition expressions should be drawn on the horizontal segments of flow charts.

Passed

All condition expressions were drawn horizontally.

Check for action transitions drawn vertically

Transitions with condition actions should be drawn on the vertical segments of flow charts.

Passed

All transitions with condition actions were drawn vertically.

Check for transition actions in flow chart

Transition actions should not be used in flow charts.

Passed

No transition actions are used in flow charts.

Check for junctions for default transitions

All Junctions in a flow chart should have a default exit transition.

Passed

All Junctions have a default exit transition.

Check for transitions that combine condition and action

Flow charts should not combine condition evaluations and action expressions in a single transition.

Passed

No combined expressions were found in the chart.

✓ Check usage of unconditional transitions in flow charts

Identify unconditional transitions in flow charts.

Passed

All unconditional transitions adhere to the guideline.

✓ Check terminal junctions in Stateflow

Identify usage of terminal junctions in flow charts.

Passed

Multiple terminal junctions were not found.

✓ Check usage of Stateflow comments

Identify comments that are nested or contain newline(s) in the middle in Stateflow for action language 'C'.

Passed

No comments found that are either nested or contain newline(s) in the middle.

🗀 *Condition Transition/Action* ✓*16* ✗*0* ⚠*0* 🗎*0*

✓ Check Stateflow chart action language

Check if the action language of Stateflow charts is set to 'C'.

Passed

All Stateflow Charts have action language set to 'C'.

✓ Check usage of numeric literals in Stateflow

Identify use of numeric literals in Stateflow states and transitions.

Passed

No numeric literals found in Stateflow charts.

✓ Check for pointers in Stateflow charts

Identify pointer operations on custom code variables.

Note: This check applies only to Stateflow charts that use C as the action language.

Passed

No pointer operations were found.

✓ Check for usage of events and broadcasting events in Stateflow charts

Identify undirected event broadcasts in Stateflow

Passed

No instances of undirected event broadcast were found.

✓ Check order of state action types

Identify out of order state action types in Stateflow states.

Passed

No Stateflow states found with out of order state action types

---

✓ Check repetition of Action types

jc_0734: Number of state action types

Identifies repeated action types in a Stateflow State.

Passed

No Stateflow States were found.

✓ Check if state action type 'exit' is used in the model

Check if state action type 'exit' is used in the model.

Passed

State action type 'exit' is not used in the model.

✓ Check updates to variables used in state transition conditions

jc_0741: Variables used in state transition conditions must not perform an update by "during" state action type.

Passed

No Stateflow states found that violate the guidelines for updating the variables used in state transition conditions.

✓ Check usage of transition conditions in Stateflow transitions

jc_0772: Identify unconditional Stateflow transitions with higher priority than conditional transitions

Passed

No unconditional Stateflow transitions found with higher priority than conditional transitions

---

✓ Check condition actions and transition actions in Stateflow

Identify usage of transition actions in Stateflow.

Passed

No Stateflow charts have transition actions.

✓ Check for MATLAB expressions in Stateflow blocks

Identify MATLAB expressions that are not suitable for code generation in Stateflow blocks.

Passed

No Stateflow objects found using MATLAB expressions unsuitable for code generation.

✓ Check usage of floating-point expressions in Stateflow charts

Identify equal to operations (==) in expressions where at least one side of the expression is a floating-point variable or constant.

Passed

No equal to operations in expressions where at least one side of the expression is a floating-point variable or constant were found.

✓ Check Stateflow operators

Identify the usage of operators in Stateflow.

Passed

No Stateflow blocks found with incorrect operator usage.

✓ Check prohibited comparison operation of logical type signals

Identify boolean variables in Stateflow charts using comparison operations.

Passed

No boolean variables use comparison operations of logical type in the model.

✓ Check usage of unary minus operations in Stateflow charts

Identify unary minus operations applied to unsigned integers in Stateflow objects.

Passed

No unary minus operations applied to unsigned integers in Stateflow objects were found.

✓ Check for implicit type casting in Stateflow

Identify implicit type casting in Stateflow.

Passed

No instances of implicit type casting found.

Label Description ✓*10* ⊗*0* ⚠*0* ▤*0*

✓ Check uniqueness of Stateflow State and Data names

jc_0732: Distinction between state name and data item name

Identify Stateflow State and Stateflow Data that have identical names in a given chart.

Passed

No Stateflow charts were found.

✓ Check uniqueness of State names

jc_0730: Independence of state name in charts

Identifies identical State names within a Stateflow Chart.

Passed

No Stateflow charts were found.

✓ Check usage of State names

jc_0731: Slash (/) in the state name

Identify state names with '/' at its end.

Passed

No Stateflow states were found.

✓ Check entry formatting in State blocks in Stateflow charts

Identify missing line breaks between entry action (en), during action (du), and exit action (ex) entries in states. Identify missing line breaks after semicolons (;) in statements.

Passed

No Stateflow charts were found.

✓ Check indentation of code in Stateflow states

Identify non-uniform indentation in Stateflow blocks.

Passed

All Stateflow blocks have uniform indentation.

✓ Check for usage of text inside states

Identify Stateflow states with text exceeding the boundary of the state

Passed

No Stateflow states found with text exceeding the boundary of the state.

✓ Check position of label string in Stateflow transition

Identify placement of label string in Stateflow transition.

Passed

All Stateflow transitions are placed uniformly.

✓ Check position of comments in transition labels

Identify comments in transition labels that are not positioned uniformly.

Passed

Comments in transition labels are positioned uniformly.

✓ Check usage of parentheses in Stateflow transitions

jc_0752: Start new line before and after parentheses for condition actions in Stateflow transitions.

Passed

No Stateflow Transitions found that violate the requirement for new line for condition actions.

✓ Check for comments in unconditional transitions

Identify comments in unconditional transitions without action statements.

Passed

All unconditional transitions without action statements have comments.

🗀 *Miscellaneous* ✓*4* ✗*0* ⚠*0* ▤*0*

✓ Check return value assignments in Stateflow graphical functions

Identify graphical functions with multiple assignments of return values in Stateflow charts.

Passed

No Stateflow charts were found.

✓ Check uniqueness of Stateflow State and Data names

jc_0732: Distinction between state name and data item name
Identify Stateflow State and Stateflow Data that have identical names in a given chart.

Passed

No Stateflow charts were found.

✓ Check usage of Simulink functions in Stateflow

Usage of Simulink Functions in Stateflow.

Passed

All Simulink Functions in Stateflow are defined according to the guideline.

✓ Check use of Simulink in Stateflow charts

na_0039: Limitation on Simulink functions in Chart blocks

Check use of Stateflow charts nested inside Simulink functions used in Stateflow.

Passed

No Stateflow charts found nested inside Simulink functions used in Stateflow.

🗀 MATLAB ✓8 ✗0 ⚠0 ▤0

🗀 *Data and Operations* *✓3 ✗0 ⚠0 ▤0*

✓ Check MATLAB code for global variables

Check for global variables in MATLAB code

Check for global variables in MATLAB code used in MATLAB Function blocks

Passed

No MATLAB Function blocks found

Check for global variables in MATLAB functions defined in Stateflow charts

Passed

No MATLAB functions defined in Stateflow charts found

Check for global variables in called MATLAB functions

Passed

No external MATLAB functions found

✓ Check usage of enumerated values

Identify enumeration classes used in the model with no default value specification.

Passed

No enumeration classes found without default value specifications.

✓ Check input and output settings of MATLAB Functions

Identify MATLAB Functions that have inputs, outputs, or parameters with inherited complexity, data type, or size properties.

Passed

No MATLAB Functions found in the model or subsystem.

Usage ✓*5* ✗*0* ⚠*0* ▤*0*

✓ Check MATLAB Function metrics

Identify MATLAB Functions that violate complexity limits.

Passed

No MATLAB Function with metrics violations were found.

The following metrics were determined for the model or subsystem.

Legend:

- LoC : Total lines of code
- ELoC : Effective lines of code
- CLoC : Comment lines of code
- DC : Density of comments
- CYC : Cyclomatic complexity

>

Input Parameters Selection

Name	Value
Maximum effective lines of code per function	60
Minimum density of comments	0.2
Maximum cyclomatic complexity per function	15

✓ Check the number of function calls in MATLAB Function blocks

Checks whether number of function calls in MATLAB Function blocks is less than 3.

Passed

Number of function calls in MATLAB Function blocks is less than 3.

✓ Check usage of character vector inside MATLAB Function block

Checks whether character vectors are being used inside MATLAB Function blocks

Passed

No character vectors found in MATLAB Function block

✓ Check usage of recommended patterns for Switch/Case statements

Checks whether non-constant variables are used in Switch/Case arguments.

Passed

Non-constant variables are not used as Switch/Case arguments

✓ Check for use of C-style comment symbols

Identify usage of C-style comments in CGT Files and MPT Objects.

Passed

C-style comments are not used in CGT Files and MPT Objects.

Modeling Standards for JMAAB ✓107 ✗0 ⚠14 ▤0

Naming Conventions ✓14 ✗0 ⚠0 ▤0

✓ *Check file names*

Check the model file name to ensure that the name complies with the recommended guidelines.

Passed

All files have correct names.

✓ *Check folder names*

Check the folder name to ensure that the name complies with the recommended guidelines.

Passed

All folders have correct names.

✓ *Check subsystem names*

Identify subsystem names with incorrect characters.

Passed

All the subsystem names use correct characters.

✓ *Check port block names*

Identify Inport or Outport block names with incorrect characters.

Passed

All the Inport or Outport block names use correct characters.

✓ *Check character usage in block names*

Identify block names with incorrect characters.

Passed

All the block names use correct characters.

✓ *Check usable characters for signal names and bus names*

Identify invalid characters in signal and bus names

Passed

No invalid characters are used in signal and bus names.

✓ *Check usable characters for parameter names*

Identify invalid characters in parameter names

Passed

No invalid characters are used in parameter names.

✓ *Check length of model file name*

Check length of model file name

Passed

Model name is valid.

✓ *Check length of folder name at every level of model path*

The model file name is: Wrong_Way_Driver_Warning

Passed

Folder names are valid.

✓ *Check length of subsystem names*

Check length of subsystem names

Passed

All subsystem names are valid.

✓ *Check length of Inport and Outport names*

Check length of Inport and Outport names

Passed

All Inport and Outport names are valid.

✓ *Check length of signal and bus names*

Check length of signal and bus names

Passed

All signal and bus names are valid.

✓ *Check length of parameter names*

Check length of parameter names

Passed

All parameter names are valid.

✓ *Check length of block names*

Check length of block names

Passed

All block names are valid.

📁 Model Architecture ✓0 ✗0 ⚠1 ▤0

⚠ *Check for mixing basic blocks and subsystems*

Identify levels in the model that include basic blocks and subsystems. Each level of a model must be designed with blocks of the same level (for example, only subsystems or only basic blocks).

Warning

The following level(s) in the model include basic blocks and subsystems:

System	**Block path**
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Ran

	ge_Conditioner/Valid_Signal/Logical Operator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Logical Operator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator1
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator2
Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal	Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator3" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Relational Operator3

Recommended Action

If possible, replace blocks at the identified level of the model hierarchy with basic blocks. Move nonvirtual blocks into the identified subsystem.

Model Configuration Options ✓2 ✖0 ⚠0 ☰0

✓ *Check Implement logic signals as Boolean data (vs. double)*

Identify whether **Implement logic signals as Boolean data (vs. double)** is selected.

Passed

Implement logic signals as Boolean data (vs. double) is selected.

✓ *Check diagnostic settings for incorrect calculation results*

Identify data validity diagnostic settings which detect incorrect calculation results.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Division by singular matrix (CheckMatrixSingularityMsg)	error	error
Pass	Inf or NaN block output (SignalInfNanChecking)	error	error
Pass	Wrap on overflow (IntegerOverflowMsg)	error	error
Pass	Saturate on overflow (IntegerSaturationMsg)	error	error

Simulink ✓40 ✖0 ⚠13 ☰0

⚠ *Check for Simulink diagrams using nonstandard display attributes*

Identify nonstandard display attributes in Simulink diagrams.

Check format settings

Identify incorrect model-level format options.

Warning

The following format display options are incorrect.

Display Attribute	Recommended Value	Actual Value
Debug > Information Overlays > Nonscalar Signals	on	off
Debug > Information Overlays > Port Data Type	off	on
Modeling > Environment > Model Browser	off	on
Debug > Information Overlays > Colors	none	disabled

Recommended Action

Set the format options to the recommended value.

Check block colors

Identify blocks using nonstandard colors.

Passed

All blocks use standard colors.

Check canvas colors

Identify canvases that are not white.

Passed

All diagrams use a white canvas.

Check diagram zoom

Identify diagrams that do not have zoom factor set to 100 %.

Note: Zoom factors can differ for each instance of a model diagram opened in Simulink Editor

Warning

The following diagrams do not have zoom factor set to 100 percent:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner

Recommended Action

For each listed diagram, select **Modeling > Environment > Zoom > Normal View (100%)**.

✓ *Check Model font settings*

Identify blocks and charts with different font settings from input parameters.

Passed

Font settings of the blocks and charts and input parameters are same.

⚠ *Check position of Inport and Outport blocks*

Check positions of Inport blocks

Warning

The following Inport blocks are not placed to left side of the diagram:

-
-

Recommended Action

Move the Inport blocks identified to the left of all other blocks in the diagram.

It is acceptable to move the Inport block to the right only to prevent signal crossings.

Check positions of Outport blocks

Warning

The following Outport blocks are not placed to right side of the diagram:

-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/WarningDisplayHMI
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SignID
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SignRelevance
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SignAboveRoad

Recommended Action

Move the Outport blocks identified to the right of all other blocks in the diagram.

It is acceptable to move the Outport block to the left only to prevent signal crossings.

✓ *Check whether block names appear below blocks*

Identify blocks where the name is not displayed below the block.

Passed

All blocks have names displayed below the block.

✓ *Check the display attributes of block names*

Identify whether to display block names.

Check for blocks with hidden names and obvious function

Identify block names that are displayed but can be hidden due to obvious behavior.

Passed

All blocks with obvious behavior have hidden names.

Check for non-descriptive displayed block names

Identify block names that are displayed but should be hidden due to a lack of a descriptive name.

Passed

All displayed names provide descriptive information.

Check for missing block names

Identify block names that are hidden but should be displayed to show a descriptive name.

Passed

All displayed names provide descriptive information.

✓ *Check for nondefault block attributes*

Identify blocks that use and fail to display nondefault values.

Passed

Model displays all block parameter values that are not default values.

✓ *Check trigger signal names*

Identify trigger blocks where the origin of the trigger signal and the destination have dissimilar names.

Passed

No violation of the guideline for use of trigger signal names.

✓ *Check for unconnected ports and signal lines*

Identify unconnected block input ports, output ports, and signal lines.

Passed

All lines and ports in the model are connected.

⚠ *Check usage of Switch blocks*

Identify Switch blocks that do not use Boolean inputs for the switch condition (input 2), and do not use `u2 ~= 0` for the **Criteria for passing first input** block parameter.

Check Switch block parameters

Identify Switch blocks with the parameter **Criteria for passing first input** not set to `u2 ~= 0`.

Warning

The block parameter **Criteria for passing first input** is not set to `u2 ~= 0` for the following blocks:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Switch
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch1" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch1
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch2" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Switch2
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Switch" title="Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Switch

Recommended Action

Set the block parameter **Criteria for passing first input** to `u2 ~= 0`. This might require reworking the logic associated with the Switch block.

____

Check for Boolean switch condition

Identify blocks that do not use Boolean signal switch conditions (input 2).

Passed

The switch condition is a Boolean signal.

____

⚠ *Check usage of Relational Operator blocks*

Identify Relational Operator blocks that connect to constants with the first (upper) input value.

Warning

The following Relational Operator blocks connect to a constant value using the first (upper) input value:

-
-
-
-

Recommended Action

Make the constant value the second (lower) input to the Relation Operator block.

____

✓ *Check Indexing Mode*

Identify blocks and charts with inconsistent Indexing mode.

Passed

No inconsistent Indexing mode used in the model.

____

✓ *Check usage of tunable parameters in blocks*

Identify tunable parameters used to specify expressions, data type conversions, or indexing operations.

Passed

Tunable parameters are not used in the model.

⚠ *Check signal line labels*

Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check source block labels

Warning

The following signals have no label:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant2/

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant3/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant4/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant5/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/Constant6/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant1/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant2/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Constant3/

Λ Less

Recommended Action

Add a new or propagated label to the signal line.

____ Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check destination block labels

Warning

The following signals have no label:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change/

Recommended Action

Add a new or propagated label to the signal line.

✓ *Check for propagated signal labels*

Identify propagated labels on signal lines.

Passed

All inputs and outputs to the subsystems and blocks have labels and display propagated signals.

✓ *Check usage of Discrete-Time Integrator block*

jc_0627: Identify Discrete-Time Integrator blocks that violate saturation limit settings

Passed

No Discrete-Time Integrator blocks found that violate JMAAB guideline jc_0627

✓ *Check settings for data ports in Multiport Switch blocks*

Identify Multiport Switch blocks that violate data port settings.

Passed

No Multiport Switch blocks found with inappropriate data port settings.

✓ *Check usage of fixed-point data type with non-zero bias*

jc_0643: Fixed-point setting

Identify blocks with a fixed-point data type whose bias is not zero.

Passed

No blocks found with the Data Type Assistant mode set to "Fixed point" and a bias value other than zero

---

✓ *Check input and output datatype for Switch blocks*

jc_0650: Identify Switch blocks with mismatched input and output data types

Passed

No Switch blocks found with mismatched input and output data types

---

✓ *Check signs of input signals in product blocks*

jc_0611: Input signal sign during product block division

Identify blocks that perform division whose inputs have different sign bit.

Passed

No product block with division of different sign bits found.

✓ *Check Signed Integer Division Rounding mode*

jc_0642: Integer rounding mode setting

Identifies blocks with block parameter 'Integer Rounding Mode' set to 'Simplest' when the configuration parameter 'Signed integer division rounds to' is set to 'Undefined'.

Passed

Configuration parameter 'Signed integer division rounds to' is not set to 'Undefined'.

⚠ *Check type setting by data objects*

jc_0644: Identify blocks that violate signal data type setting if signal objects are used.

Warning

The following blocks violate signal data type setting if signal objects are used.

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData/Switch
-
-

Recommended Action

Set the output data type of the blocks either to "auto" or "Inherit via back propagation". This check excludes Data Type Conversion block, type setting by fixdt, double and boolean data types, and reusable internal part of function (treat as atomic unit).

✓ *Check usage of the Saturation blocks*

jc_0628: Identify the Saturation and Saturation Dynamic blocks that perform type casting.

Passed

No Saturation and/or Saturation Dynamic blocks perform type casting

---

✓ *Check usage of Merge block*

jc_0659: Usage restrictions of signal lines inputted to Merge block
There must not be any block between a Conditional Subsystem block and a Merge block.

Passed

No Merge block found.

✓ *Check usage of Memory and Unit Delay blocks*

Identify Memory blocks not using a continuous sample time

Passed

No Memory blocks found with inappropriate sample time

Identify Unit Delay blocks with non-discrete sample time

Passed

No Unit Delay blocks found with non-discrete sample time

✓ *Check block orientation*

Identify blocks which are rotated or reversed

Passed

No blocks found with rotated or reversed orientation

✓ *Check if blocks are shaded in the model*

Check if blocks are shaded in the model

Passed

Blocks in the model are not shaded.

✓ *Check operator order of Product blocks*

Operator order for Product blocks.

Passed

All Product blocks have valid operator order.

✓ *Check icon shape of Logical Operator blocks*

Icon shape of Logical Operator blocks

Passed

All Logical Operator blocks have consistent icon shape.

⚠ *Check if tunable block parameters are defined as named constants*

Check if tunable block parameters are defined as named constants

Warning

The following tunable block parameters are not defined as named constants.

Block	Violations
	Value : -60
	Value : 60
	Value : 255

Recommended Action

Consider changing tunable block parameter literal values to named constants.

✓ *Check default/else case in Switch Case blocks and If blocks*

Check if default/else case in Switch Case blocks and If blocks are set to 'on'

Passed

Conditional Control blocks are valid.

✓ *Check usage of Lookup Tables*

jc_0626: Guideline for using the Lookup Table system block
Checks for the recommended parameter settings in Lookup Tables to prevent unexpected results.

Passed

All the Lookup Tables pass the check.

✓ *Check for parentheses in Fcn block expressions*

jc_0622: Guideline for using the Fcn block

Passed

All Fcn blocks use parentheses to mark operator precedence.

✓ *Check undefined initial output for conditional subsystems*

Check undefined initial output for Outports/Merge blocks in conditional subsystems

Passed

The initial output setting for all Conditional Subsystems are valid.

✓ *Check for avoiding algebraic loops between subsystems*

jc_0653: Guidelines for avoiding algebraic loops between subsystems.

Passed

No delay blocks in feedback loops violate the guidelines for avoiding algebraic loops between subsystems.

✓ *Comparing floating point types in Simulink*

jc_0800: Comparing floating point types in Simulink
Equivalence comparison should not be used for floating point numbers.

Passed

No Equivalence comparison done on floating point numbers.

✓ *Check duplication of Simulink Data names*

Simulink Data names should be unique across base workspace, model workspace and data dictionary.

Passed

All Simulink Data names are unique.

⚠ *Check unused data in Simulink Model*

Check for unused data in Data Dictionary

Warning

The following data variables in the data dictionary are unused:

Data Objects	**Source**
F32	WWDW_dd.sldd
F64	WWDW_dd.sldd
FLAG	WWDW_dd.sldd
MODE	WWDW_dd.sldd
S16	WWDW_dd.sldd
S32	WWDW_dd.sldd
S8	WWDW_dd.sldd
SMODE	WWDW_dd.sldd
SignPosLat	WWDW_dd.sldd
Sign_Status	WWDW_dd.sldd

U16	WWDW_dd.sldd
U32	WWDW_dd.sldd
U8	WWDW_dd.sldd
ValidLateralPosition	WWDW_dd.sldd
ValidLongitudionalPosition	WWDW_dd.sldd
VehicleSpeed_kph	WWDW_dd.sldd
VehicleYawRate_Degree	WWDW_dd.sldd

∧ Less

Recommended Action

Consider removing the unused data variables.

⚠ *Check output data type of operation blocks*

jc_0651: Guideline for implementing a type conversion.

Warning

Following operation blocks explicitly specify output data type:

-
-
-
-
-
-
-

-
-
-
-
-
-
-
-
-
-
-

∧ Less

Recommended Action

Instead of explicitly specifying output data type on operation blocks, use 'Data Type Conversion' block when changing the data type of the block output signal.

⚠ *Check Model Description*

Identify layers in the model having inconsistent description format.

Warning

Following layers do not have model descriptions:

- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData

-
-
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
-
-
-
-
-

ʌ Less

Recommended Action

Consider adding model description for all the layers.

✓ *Check for consistency in model element names*

Check if model elements connected to a signal are following consistent naming.

Passed

Model elements connected to a signal are following consistent names.

✓ *Check for sample time setting*

Check if sample time property of a block is set to -1 (inherited).

Passed

All permitted blocks have sample time set to -1 (inherited).

✓ *Check usage of Sum blocks*

Identify inappropriate usages of Sum block.

Passed

No violations of the guideline found with the usage of the Sum block.

⚠ *Check position of signal labels*

Check location of signal labels

Warning

The following signals do not have labels located at the origin of the signal line:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Camera_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/EnabledData
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Map_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/SuppressedOutput
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Lateral_Sign_Value_in_ISO_Coordinates
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal/Sign_Positional_Longitudional
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Valid_Signal
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Data_Conditioning
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Vehicle_Speed_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Yaw_Rate_Conversion
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning

- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning
- Wrong_Way_Driver_Warning

^ Less

Recommended Action

Consider placing the labels at the origin of the signal line.

✓ *Check for missing ports in Variant Subsystems*

Check for number of inputs/outputs to a Variant Subsystem.

Passed

No Variant Subsystems found having different number of inputs/outputs in the Variant Subsystem choices.

✓ *Check for cascaded Unit Delay blocks*

Identify cascaded and tapped pattern of Unit Delay blocks.

Passed

No cascaded Unit Delay blocks found that can be changed to Tapped Delay/Delay block.

✓ *Check for usage of Data Store Memory blocks*

Identify the usage of Data Store Memory blocks.

Passed

Usage of Data Store Memory blocks is correct.

⚠ *Check fundamental logical and numerical operations*

Check input data types of blocks meant for numerical operations

Warning

The following numerical operation blocks have boolean data type as input:

-
-
-
-
-
-
-
-
-

Recommended Action

Consider having non-boolean inputs for the numerical operation blocks.

✓ *Check signal flow in model*

Identify subsystems which do not have a signal flow from left to right.

Passed

No subsystems found with inappropriate signal flow.

✓ *Check usage of vector and bus signals*

Identify mixed usages of vector and bus signals.

Passed

No mixing of vector and bus signals found in the system.

✓ *Check connections between structural subsystems*

Identify connections between structural subsystems.

Passed

All connections to structural subsystems adhere to the guideline.

✓ *Check position of conditional blocks and iterator blocks*

Identify conditional and iterative blocks that are positioned inconsistently in the model.

Passed

The conditional and iterative blocks are correctly placed in the model.

⚠ *Check signal line connections*

Check signal intersections

Warning

The following signals intersect with other signals in the diagram:

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change

- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change
- Wrong_Way_Driver_Warning/Input_Signal_Range_Conditioner/Input_Signal_Range_Change

Recommended Action

Reposition the above listed signals to avoid intersections.

✓ *Check scope of From and Goto blocks*

Identify incorrect scoping of From and Goto blocks. For signal flows, From and Goto blocks must use local scope. Control flow can use global scope.

Passed

All From and Goto blocks are used correctly.

🗀 Stateflow ✓49 ✖0 ⚠0 ☰0

✓ *Check transition orientations in flow charts*

Identify transitions in Stateflow flow charts that are drawn incorrectly.

Check for conditions drawn horizontally

Condition expressions should be drawn on the horizontal segments of flow charts.

Passed

All condition expressions were drawn horizontally.

Check for action transitions drawn vertically

Transitions with condition actions should be drawn on the vertical segments of flow charts.

Passed

All transitions with condition actions were drawn vertically.

Check for transition actions in flow chart

Transition actions should not be used in flow charts.

Passed

No transition actions are used in flow charts.

Check for junctions for default transitions

All Junctions in a flow chart should have a default exit transition.

Passed

All Junctions have a default exit transition.

Check for transitions that combine condition and action

Flow charts should not combine condition evaluations and action expressions in a single transition.

Passed

No combined expressions were found in the chart.

✓ *Check return value assignments in Stateflow graphical functions*

Identify graphical functions with multiple assignments of return values in Stateflow charts.

Passed

No Stateflow charts were found.

✓ *Check default transition placement in Stateflow charts*

jc_0531: Placement of default transition.

Passed

No Stateflow transitions and states found that violate the guidelines for default transition placement in Stateflow charts.

✓ *Check for Strong Data Typing with Simulink I/O*

Check whether labeled input and output signals are strongly typed.

Passed

No Stateflow charts have **Use Strong Data Typing with Simulink I/O** cleared.

✓ *Check definition of Stateflow data*

Identify the Scope value set on Stateflow data defined at machine level.

Passed

All Stateflow data at machine level has been defined as per guideline.

✓ *Check for MATLAB expressions in Stateflow blocks*

Identify MATLAB expressions that are not suitable for code generation in Stateflow blocks.

Passed

No Stateflow objects found using MATLAB expressions unsuitable for code generation.

✓ *Check for pointers in Stateflow charts*

Identify pointer operations on custom code variables.

Note: This check applies only to Stateflow charts that use C as the action language.

Passed

No pointer operations were found.

✓ *Check Stateflow operators*

Identify the usage of operators in Stateflow.

Passed

No Stateflow blocks found with incorrect operator usage.

✓ *Check usage of unary minus operations in Stateflow charts*

Identify unary minus operations applied to unsigned integers in Stateflow objects.

Passed

No unary minus operations applied to unsigned integers in Stateflow objects were found.

✓ *Check usage of Stateflow comments*

Identify comments that are nested or contain newline(s) in the middle in Stateflow for action language 'C'.

Passed

No comments found that are either nested or contain newline(s) in the middle.

✓ *Check prohibited comparison operation of logical type signals*

Identify boolean variables in Stateflow charts using comparison operations.

Passed

No boolean variables use comparison operations of logical type in the model.

✓ *Check usage of internal transitions in Stateflow states*

Identify Stateflow states using multiple internal transitions.

Passed

No Stateflow states found with multiple internal transitions

✓ *Check usage of transition conditions in Stateflow transitions*

jc_0772: Identify unconditional Stateflow transitions with higher priority than conditional transitions

Passed

No unconditional Stateflow transitions found with higher priority than conditional transitions

---

✓ *Check uniqueness of Stateflow State and Data names*

jc_0732: Distinction between state name and data item name
Identify Stateflow State and Stateflow Data that have identical names in a given chart.

Passed

No Stateflow charts were found.

✓ *Check uniqueness of State names*

jc_0730: Independence of state name in charts
Identifies identical State names within a Stateflow Chart.

Passed

No Stateflow charts were found.

✓ *Check usage of parentheses in Stateflow transitions*

jc_0752: Start new line before and after parentheses for condition actions in Stateflow transitions.

Passed

No Stateflow Transitions found that violate the requirement for new line for condition actions.

✓ *Check prohibited combination of state action and flow chart*

jc_0762: State actions within states and flow chart statements should not be used in combination.

Passed

No Stateflow states found that combine state action and flow chart.

✓ *Check condition actions and transition actions in Stateflow*

Identify usage of transition actions in Stateflow.

Passed

No Stateflow charts have transition actions.

✓ *Check usable number for first index*

Identify usage of first index of Stateflow data.

Passed

All Stateflow data first index values are uniform.

✓ *Check usage of State names*

jc_0731: Slash (/) in the state name
Identify state names with '/' at its end.

Passed

No Stateflow states were found.

✓ *Check execution timing for default transition path*

'Execute (enter) Chart At Initialization' should be set to OFF.

Passed

All Stateflow Charts pass the check.

✓ *Check repetition of Action types*

jc_0734: Number of state action types
Identifies repeated action types in a Stateflow State.

Passed

No Stateflow States were found.

✓ *Check for unused data in Stateflow Charts*

Checks if the model parameter 'Unused data, events, messages and functions' is not set to 'none'.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values
Pass	Unused data, events, messages and functions (SFUnusedDataAndEventsDiag)	warning	error, warning

---

✓ *Check updates to variables used in state transition conditions*

jc_0741: Variables used in state transition conditions must not perform an update by "during" state action type.

Passed

No Stateflow states found that violate the guidelines for updating the variables used in state transition conditions.

✓ *Check starting point of internal transition in Stateflow*

jc_0760: In all state charts and flow charts, internal transitions from state boundaries must start from the left edge of the state.

Passed

No Stateflow transitions found that violate the guidelines for starting point of internal transition in Stateflow.

✓ *Check for parallel Stateflow state used for grouping*

jc_0721: Guidelines for using parallel states
Identify parallel Stateflow States used for grouping.

Passed

No Stateflow charts were found.

✓ *Check scope of data in parallel states*

jc_0722: Guidelines for setting local variables in parallel states

The scope of local variables should be restricted to one parallel state unless it is being used by other parallel states.

Passed

No Stateflow States were found.

✓ *Check indentation of code in Stateflow states*

Identify non-uniform indentation in Stateflow blocks.

Passed

All Stateflow blocks have uniform indentation.

✓ *Check for usage of text inside states*

Identify Stateflow states with text exceeding the boundary of the state

Passed

No Stateflow states found with text exceeding the boundary of the state.

✓ *Check for unexpected backtracking in state transitions*

Identify configuration parameter settings which identify unexpected backtracking in state transitions.

Passed

All constraints on model configuration parameters have been met.

Status	**Parameter**	**Current Value**	**Recommended Values**
Pass	Unexpected backtracking (SFUnexpectedBacktrackingDiag)	error	error

✓ *Check for unconnected objects in Stateflow Charts*

Identify dangling transitions and unconnected Stateflow States and Junctions in Stateflow Charts.

Passed

No unconnected transitions, states or junctions found in Stateflow Charts.

✓ *Check position of label string in Stateflow transition*

Identify placement of label string in Stateflow transition.

Passed

All Stateflow transitions are placed uniformly.

✓ *Check Stateflow chart action language*

Check if the action language of Stateflow charts is set to 'C'.

Passed

All Stateflow Charts have action language set to 'C'.

✓ *Check usable characters for Stateflow data names*

Identify invalid characters in Stateflow data names.

Passed

No invalid characters are used in Stateflow data names.

✓ *Check length of Stateflow data name*

Check if the length of Stateflow data names are within limit.

Passed

All Stateflow data names are valid.

✓ *Check usage of transitions to external states*

Identify transitions ending on external child states.

Passed

No direct transitions found from external state to child state.

✓ *Check order of state action types*

Identify out of order state action types in Stateflow states.

Passed

No Stateflow states found with out of order state action types

---

✓ *Check usage of numeric literals in Stateflow*

Identify use of numeric literals in Stateflow states and transitions.

Passed

No numeric literals found in Stateflow charts.

✓ *Check position of comments in transition labels*

Identify comments in transition labels that are not positioned uniformly.

Passed

Comments in transition labels are positioned uniformly.

✓ *Check terminal junctions in Stateflow*

Identify usage of terminal junctions in flow charts.

Passed

Multiple terminal junctions were not found.

✓ *Check for implicit type casting in Stateflow*

Identify implicit type casting in Stateflow.

Passed

No instances of implicit type casting found.

✓ *Check if state action type 'exit' is used in the model*

Check if state action type 'exit' is used in the model.

Passed

State action type 'exit' is not used in the model.

✓ *Check for use of C-style comment symbols*

Identify usage of C-style comments in CGT Files and MPT Objects.

Passed

C-style comments are not used in CGT Files and MPT Objects.

✓ *Check usage of unconditional transitions in flow charts*

Identify unconditional transitions in flow charts.

Passed

All unconditional transitions adhere to the guideline.

✓ *Check for comments in unconditional transitions*

Identify comments in unconditional transitions without action statements.

Passed

All unconditional transitions without action statements have comments.

✓ *Check definition of Stateflow events*

Stateflow events should be defined at the smallest possible scope of usage.

Passed

All Stateflow events are defined at their smallest scope.

✓ *Check Stateflow transition appearance*

Identify Stateflow transitions visually overlapping other Stateflow objects.

Passed

No transition violates the guidelines for Stateflow transition appearance.

✓ *Check for usage of events and broadcasting events in Stateflow charts*

Identify undirected event broadcasts in Stateflow

Passed

No instances of undirected event broadcast were found.

✓ *Check usage of Simulink functions in Stateflow*

Usage of Simulink Functions in Stateflow.

Passed

All Simulink Functions in Stateflow are defined according to the guideline.

📁 MATLAB Functions ✓2 ✗0 ⚠0 ▤0

✓ *Check input and output settings of MATLAB Functions*

Identify MATLAB Functions that have inputs, outputs, or parameters with inherited complexity, data type, or size properties.

Passed

No MATLAB Functions found in the model or subsystem.

✓ *Check MATLAB code for global variables*

Check for global variables in MATLAB code

Check for global variables in MATLAB code used in MATLAB Function blocks

Passed

No MATLAB Function blocks found

Check for global variables in MATLAB functions defined in Stateflow charts

Passed

No MATLAB functions defined in Stateflow charts found

Check for global variables in called MATLAB functions

Passed

No external MATLAB functions found

Units Inconsistencies ✓0 ✗0 ⚠0 ☰5

☰ Identify unit mismatches in the model

Not Run

☰ Identify automatic unit conversions in the model

Not Run

☰ Identify disallowed unit systems in the model

Not Run

☰ Identify undefined units in the model

Not Run

☰ Identify ambiguous units in the model

Not Run

Upgrading to the Current Simulink Version ✓0 ✗0 ⚠0 ☰1

☰ Open the Upgrade Advisor

Not Run

Modeling Standards for MISRA C:2012 ✓13 ✗0 ⚠0 ☰0

✓ Check configuration parameters for MISRA C:2012

Identify configuration parameters that might impact MISRA C:2012 compliant code generation.

Passed

All constraints on model configuration parameters have been met.

Status	Parameter	Current Value	Recommended Values	Prerequisites
Pass	Model Verification block enabling (AssertControl)	DisableAll	DisableAll	
D - Pass	Shared code placement (UtilityFuncGeneration)	Shared location	Shared location	
Pass	Generate shared constants (GenerateSharedConstants)	off	off	UtilityFuncGeneration
D - Pass	System target file (SystemTargetFile)	ERT based target	ERT based target	
Pass	continuous time (SupportContinuousTime)	off	off	SystemTargetFile
Pass	non-inlined S-functions (SupportNonInlinedSFcns)	off	off	SystemTargetFile
Pass	MAT-file logging (MatFileLogging)	off	off	
Pass	Code replacement library (CodeReplacementLibrary)	None	None, AUTOSAR 4.0	
Pass	Parentheses level (ParenthesesLevel)	Maximum	Maximum	SystemTargetFile

Pass	Casting modes (CastingMode)	Standards	Standards	SystemTargetFile
Pass	System-generated identifiers (InternalIdentifier)	Shortened	Shortened	SystemTargetFile
Pass	Signed integer division rounds to (ProdIntDivRoundTo)	Zero	Zero, Floor	
Pass	Use division for fixed-point net slope computation (UseDivisionForNetSlopeComputation)	UseDivisionForReciprocalsOfIntegersOnly	on, UseDivisionForReciprocalsOfIntegersOnly	
Pass	Replace multiplications by powers of two with signed bitwise shifts (EnableSignedLeftShifts)	off	off	SystemTargetFile
Pass	Allow right shifts on signed integers (EnableSignedRightShifts)	off	off	SystemTargetFile
Pass	Wrap on overflow (IntegerOverflowMsg)	error	warning, error	
Pass	Inf or NaN block output (SignalInfNanChecking)	error	warning, error	
Pass	Dynamic memory allocation in MATLAB functions (MATLABDynamicMemAlloc)	off	off	
Pass	External mode (ExtMode)	off	off	
Pas	Undirected event broadcasts	error	error	

s	(SFUndirectedBroadcastEventsDiag)			
Pass	Compile-time recursion limit for MATLAB functions (CompileTimeRecursionLimit)	0	0	
Pass	Enable run-time recursion for MATLAB functions (EnableRuntimeRecursion)	off	off	
D - Pass	Include comments (GenerateComments)	on	on	
Pass	MATLAB user comments (MATLABFcnDesc)	on	on	GenerateComments, SystemTargetFile

∧ Less

✓ Check for blocks not recommended for C/C++ production code deployment

Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

Passed
Blocks not recommended for C/C++ production code deployment were not found in the model or subsystem.

✓ Check for blocks not recommended for MISRA C:2012

Identify blocks that are not recommended for MISRA C:2012 compliant code generation.

Passed

None of the blocks are defined as "not recommended" for MISRA C:2012 compliant code generation.

✓ Check for unsupported block names

Identify block names containing "/".

Passed

No unsupported block names found.

✓ Check usage of Assignment blocks

Identify Assignment blocks with possibly incomplete array initialization that do not have the simulation run-time diagnostic **Action if any output element is not assigned** set to:

- Warning, if Assignment block is in an iterator subsystem
- Error, if Assignment block is not in an iterator subsystem

Passed

All Assignment blocks are configured with block parameter **Action if any output element is not assigned** set to Warning or Error.

✓ Check for switch case expressions without a default case

Identify switch case expressions that do not have a default case.

Passed

All switch case expressions have default cases.

✓ Check for missing error ports in AUTOSAR receiver interfaces

Identify AUTOSAR receiver interface ports that do not have a matching error port.

Passed

Model is not configured as an AUTOSAR target.

✓ Check for bitwise operations on signed integers

Identify bitwise operations on signed integers.

Passed

No bitwise operations on signed integers found.

✓ Check for recursive function calls

Identify function calls that are recursive.

Passed

No recursive function calls found.

✓ Check for equality and inequality operations on floating-point values

Identify equality and inequality operations on floating-point values.

Passed

No equality or inequality operations on floating-point values found.

✓ Check for missing const qualifiers in model functions

Identify missing const qualifiers in model functions.

Passed

Model does not use customized model functions.

✓ Check integer word lengths

Identify integer word length that are not compliant with hardware implementation settings.

Passed

All used integer word length are compliant with hardware implementation settings.

✓ Check bus object names that are used as bus element names

Identify bus object names that are used as bus element names.

Passed

No bus object names are used as bus element names.

Modeling Standards for Secure Coding (CERT C, CWE, ISO/IEC TS 17961) ✓0 ✗0 ⚠0 ▤13

Check configuration parameters for secure coding standards

Not Run

Check for blocks not recommended for C/C++ production code deployment

Not Run

Check for blocks not recommended for secure coding standards

Not Run

Check usage of Assignment blocks

Not Run

Check for switch case expressions without a default case

Not Run

Check for bitwise operations on signed integers

Not Run

Check for equality and inequality operations on floating-point values

Not Run

Check integer word lengths

Not Run

Detect Dead Logic

Not Run

Detect Integer Overflow

Not Run

Detect Division By Zero

Not Run

Detect Out Of Bound Array Access

Not Run

Detect Specified Minimum and Maximum Value Violations

Not Run

Frequency Response Estimation ✓0 ✗0 ⚠0 ▤1

Identify time-varying source blocks interfering with frequency response estimation

Not Run