

GOOD INTERFACES IN LARGE MODELS

Heiko Doerr

MGI Group, March 07, 2017

LÄUFT DIE SOFTWARE,
FÄHRT DAS AUTO.

SOLUTIONS FOR INTEGRATED QUALITY ASSURANCE OF
EMBEDDED AUTOMOTIVE SOFTWARE



- Functions grow with increasing complexity
- Integration of data leads to added-value applications

- Interfaces of functions become larger and larger
 - More sensor data
 - Higher integration with other functions
 - More actuators

- Challenge: Control and manage amount of input and output variables by definition of appropriate architecture
 - How to deal with interfaces with $\gg 100$ signals?

Principles for software architectural design	ASIL				Realization in MbD
	A	B	C	D	
1a) Hierarchical structure of software components	++	++	++	++	<ul style="list-style-type: none"> • Definition of a suitable subsystem hierarchy • Checkable with MXAM/M-XRAY
1b) Restricted size of software components	++	++	++	++	<ul style="list-style-type: none"> • Suitable partitioning of functionality • Checkable with MXAM/M-XRAY
1c) Restricted size of interfaces	+	+	+	+	<ul style="list-style-type: none"> • Definition of a suitable bus hierarchy • Checkable with MXAM/M-XRAY

ISO26262-6 Table 3 - Principles for software architectural design

Principles for software architectural design	ASIL				Realization in MbD
	A	B	C	D	
1d) High cohesion within each software component	+	++	++	++	<ul style="list-style-type: none"> • Suitable partitioning of functionality • Requires manual review
1e) Restricted coupling between software components	+	++	++	++	<ul style="list-style-type: none"> • Suitable partitioning of functionality • Requires manual review
1f) Appropriate scheduling properties	++	++	++	++	<ul style="list-style-type: none"> • Timing requirements met • Semiautomatic assessment
1g) Restricted use of interrupts	+	+	+	++	<ul style="list-style-type: none"> • Not relevant for MbD

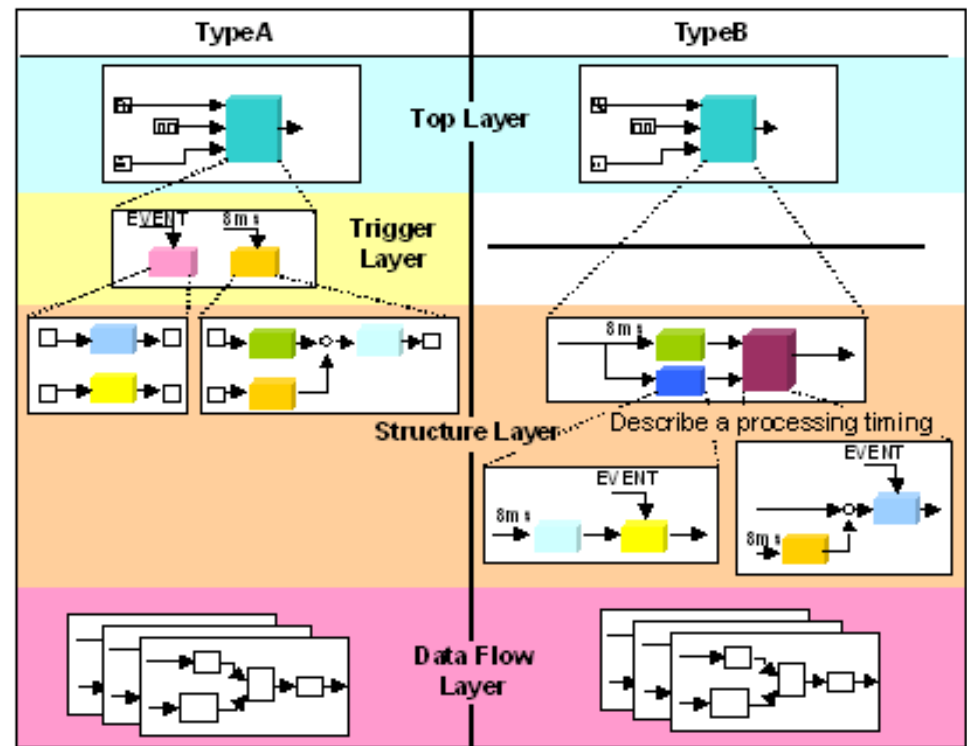
ISO26262-6 Table 3 - Principles for software architectural design

□ jc_o301: *Controller Model*

Control models are organized using the following hierarchical structure:

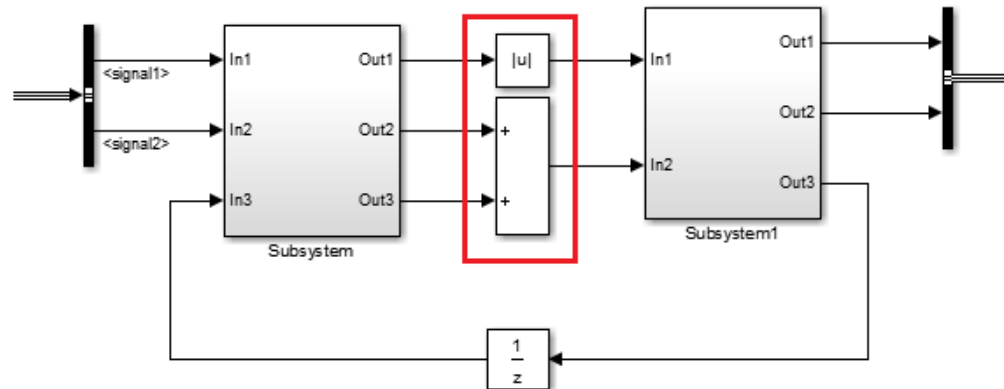
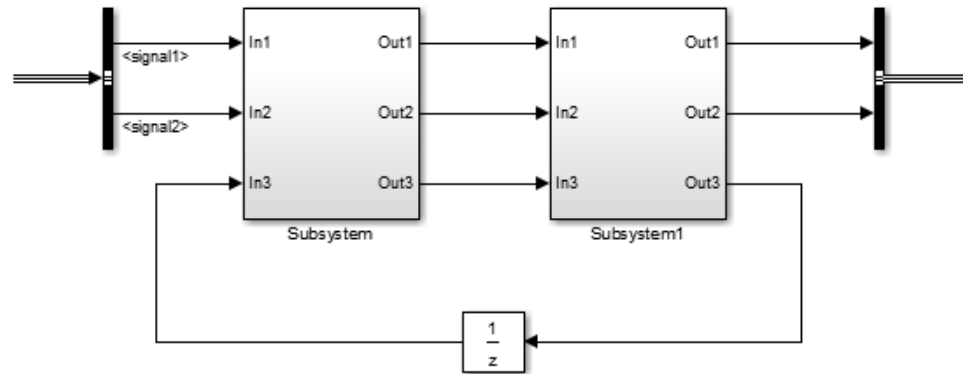
- Top layer / root level
- Trigger layer
- Structure layer
- Data Flow layer

- Use structure layer(s) to distribute signals between subsystems



□ db_o143: *Similar Block Types on the Model Levels*

- At Structure layer use subsystems and busses
- Do not mix structure and computation / algorithm



□ na_0010: Grouping Data Flows Into Signals

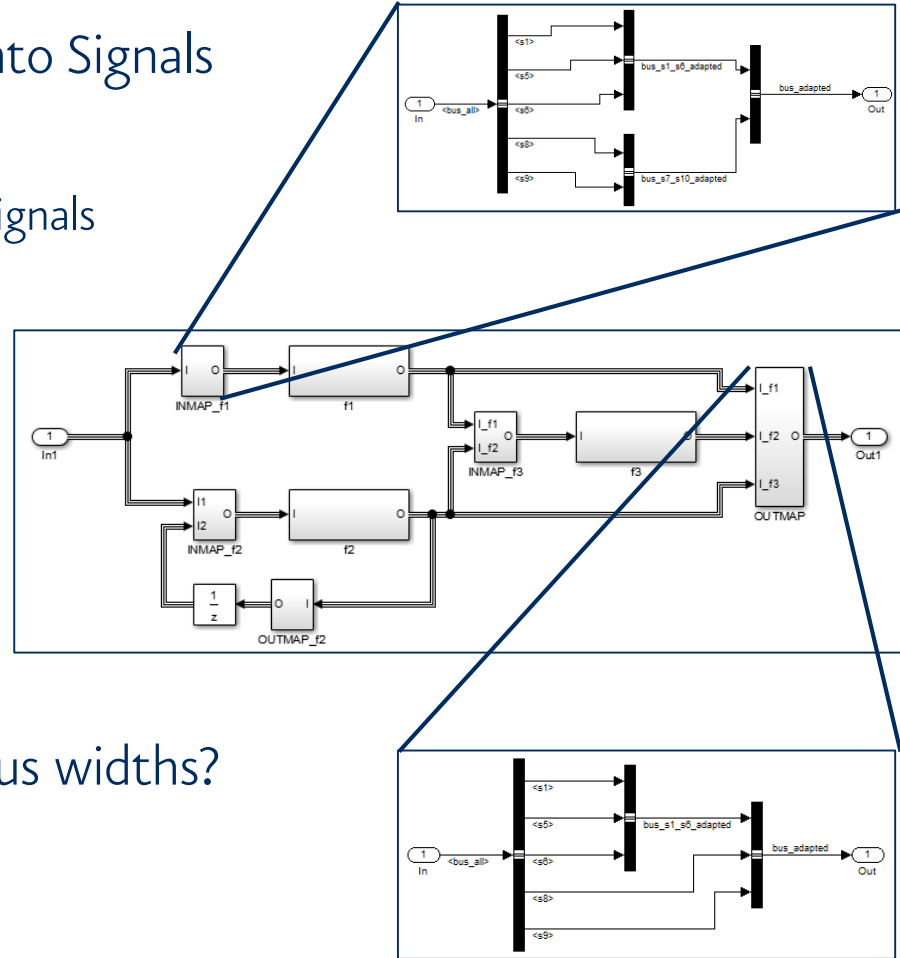
- sic! "... Into Signals"
- Busses shall serve as containers of signals

□ Improved readability

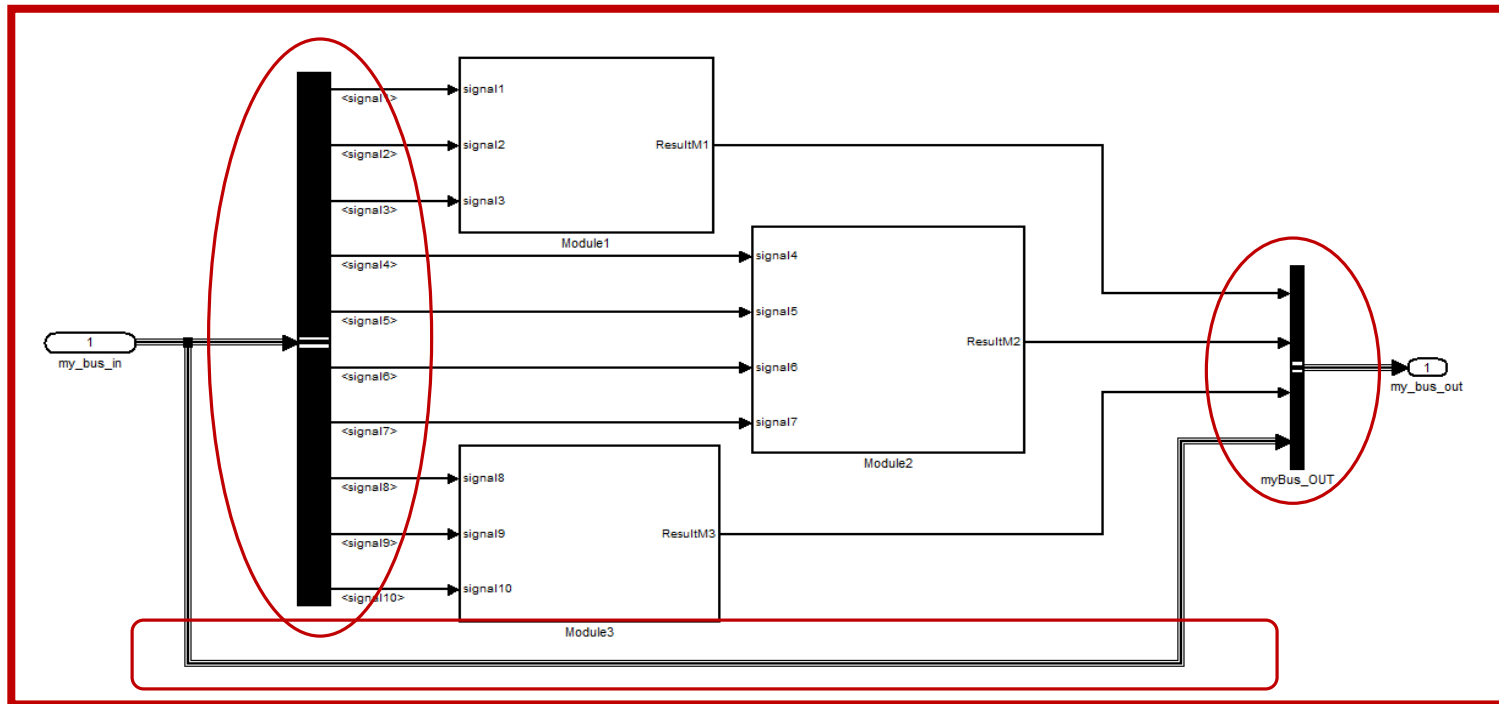
- Adaptation of bus signals in dedicated subsystems
- Adaptation to interface of subsystem



Are there any constraints on bus widths?

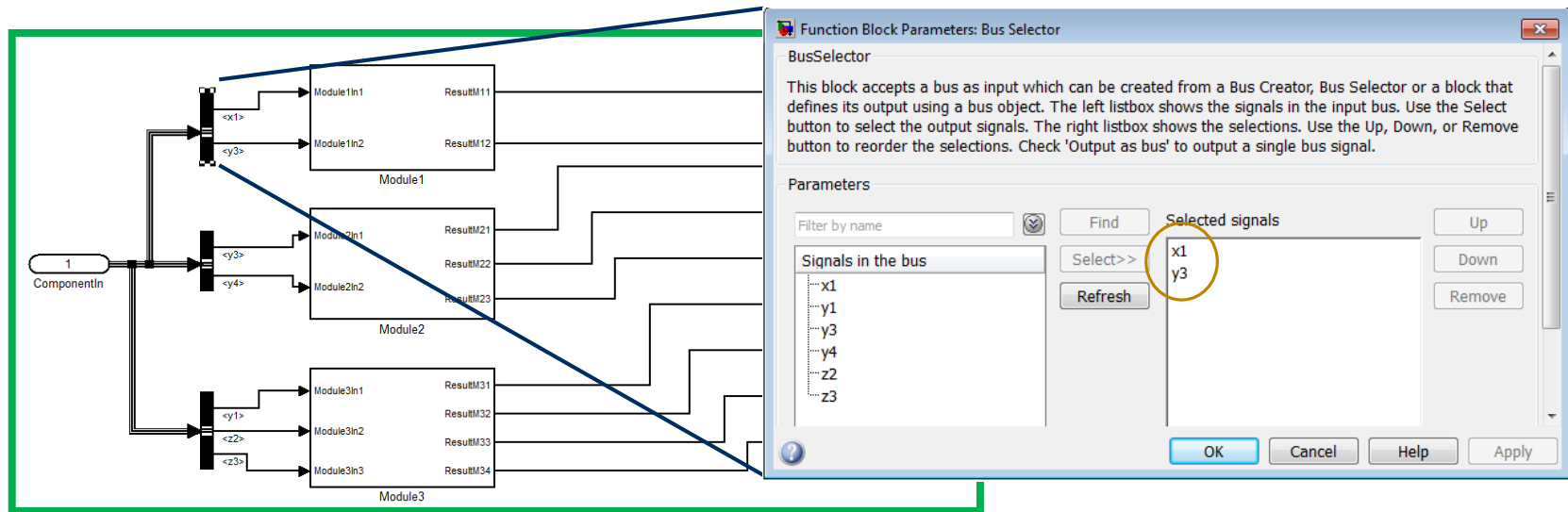


- General restriction of number of inports/outports not reasonable
- „effective“ interfaces
- Suitable definition of bus hierarchy



ISO26262-6 Table 3, 1c - Principles for software architectural design

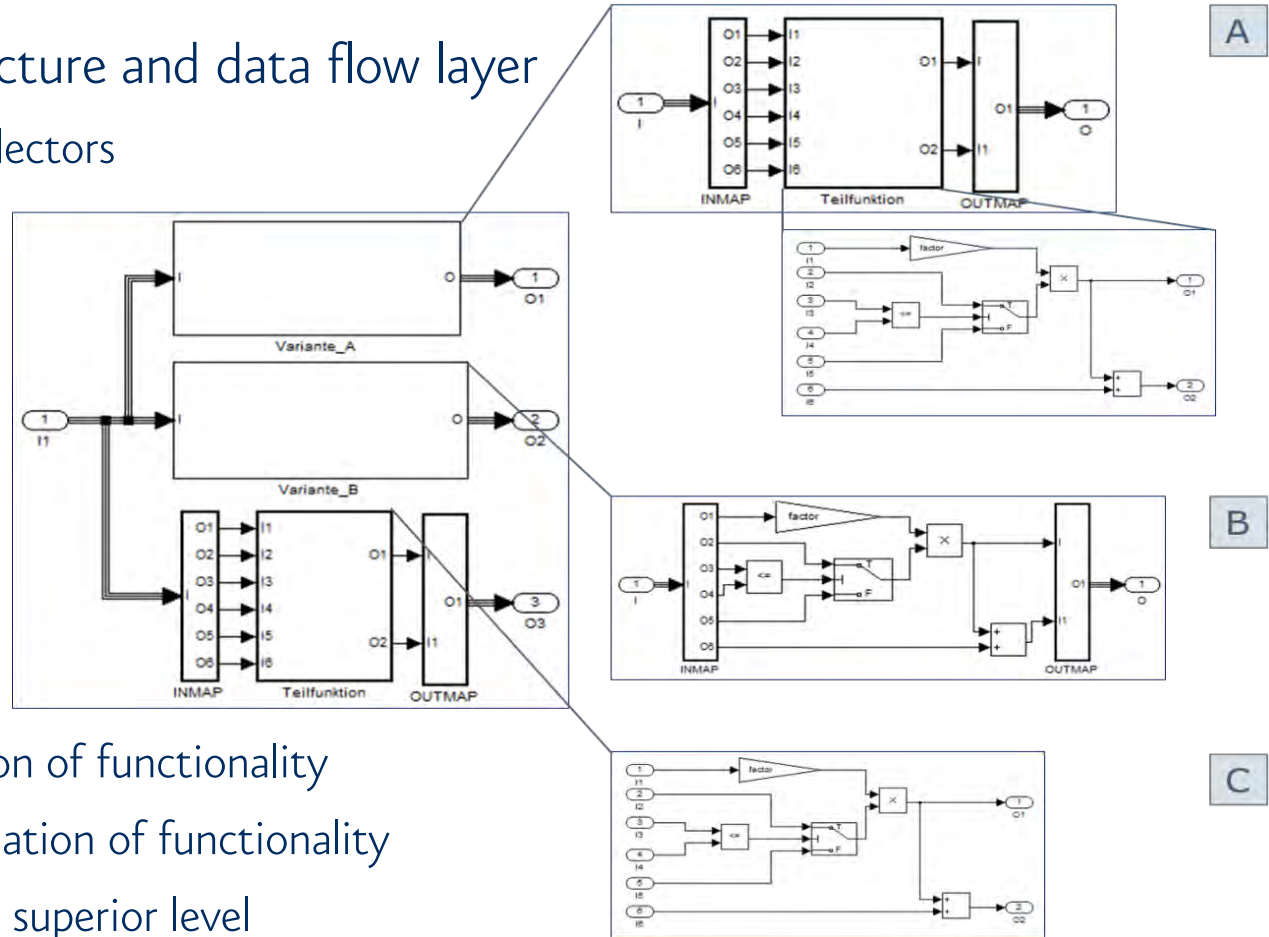
- ❑ Interface shall only contain signals used in subsystem
- ❑ Extract or adapt bus signals outside the subsystem
- ❑ Benefit: Improve readability and testability



How to enforce definition of effective interfaces?

□ Border between structure and data flow layer

- Shall also use bus selectors

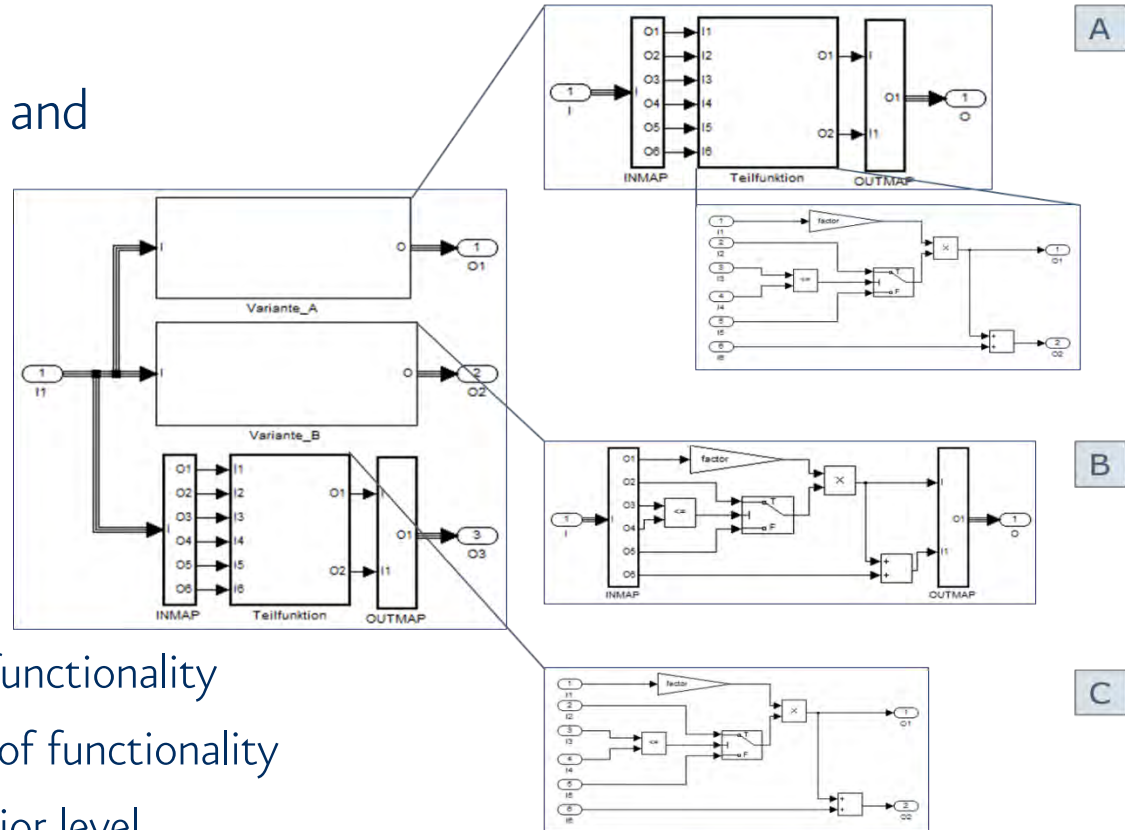


□ Design options for Inmap/Outmap

- A: With encapsulation of functionality
- B: Without encapsulation of functionality
- C: decomposition at superior level

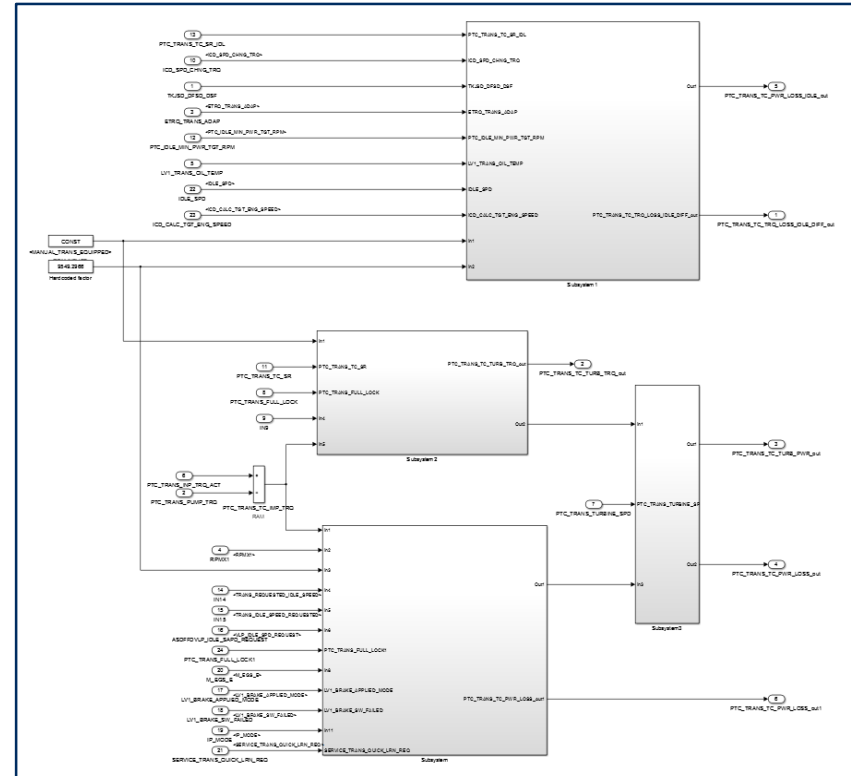
- Border between structure and data flow layer
 - Shall also use bus selectors

- Design options for Inmap/Outmap
 - A: With encapsulation of functionality
 - B: Without encapsulation of functionality
 - C: decomposition at superior level



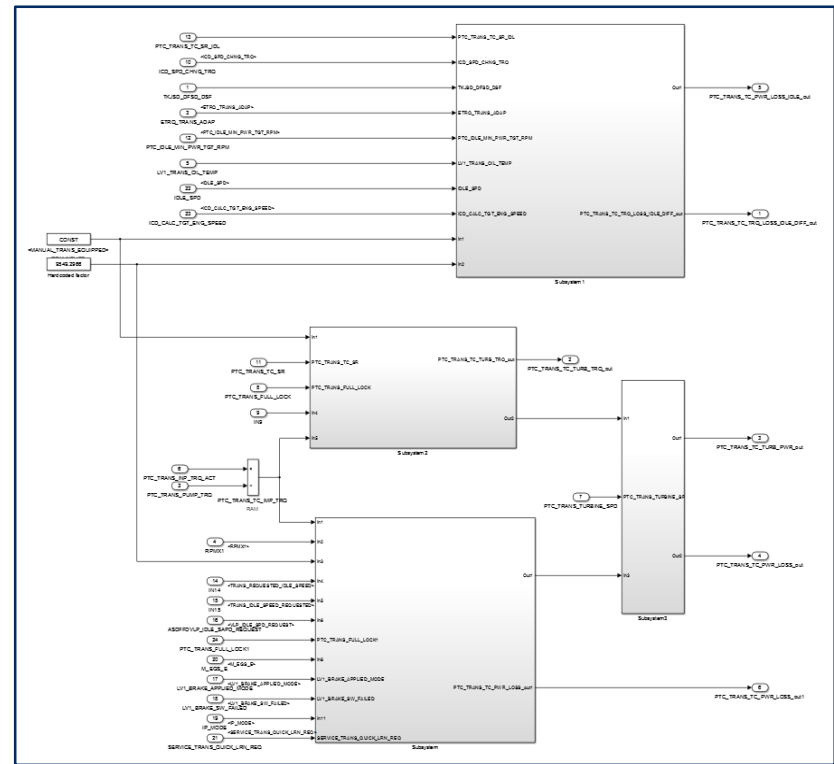
Any preferences for options ?

- ❑ Decomposition of busses reveals all signals available for computation
- ❑ Benefit: data flow fully visible
- ❑ Proposal: Avoid busses in subsystems at data flow layer
 - If number of signals is “too high” add further decomposition at structure layer



Do you agree to proposal ?

- Decomposition of busses reveals all signals available for computation
 - Consequence: Size of system interface may become large
- Specific concerns:
 - ✦ Do you have specific guidelines to determine the width of interfaces?
 - ✦ If yes, how do these guidelines look like?
 - ✦ If yes, are there model patterns by which this widths can be obtained?





Further input to the topic ?

□ **Tuesday, June 20, 2017**

3:00 pm CET (Berlin)

9:00 am EDT (Detroit)

6:30 pm IST (Bangalore)

09:00 pm CST (Beijing)

10:00 pm JST (Tokyo)



□ **Link to Event:**

<https://model-engineers-event-en.webex.com/model-engineers-event/onstage/g.php?MTID=ef3fb308217c49d54910865af8a8d96e2>



MODEL ENGINEERING SOLUTIONS GMBH

Mauerstraße 79
10117 Berlin
Germany

T: +49 30 2091 6463-0
F: +49 30 2091 6463-33

info@model-engineers.com
www.model-engineers.com

