Accellera Functional Safety Working Group
Update and Next Steps

Alessandra Nardi, Accellera FS WG Chair
Ghani Kanawati, Technical Director of System Reliability, ARM
Agenda

• The Accellera Functional Safety Working Group (FS WG)
  • Mission and the FS Standardization Landscape
  • Scope and Key Objectives
• The Accellera Functional Safety Standard
  • Data Model Development
  • Data Model White Paper
  • An example
• What’s Next?
Mission and Key Objectives
Mission of the FS WG

1. Exchange of same FS data across different automation tools
2. Connection between the FS data and the design information
3. Sharing of different FS data across operations/work products in the same layer
4. Exchange of FS data between suppliers and integrators
5. Traceability of information across the distributed development environment

- Define a FS language to capture and propagate the functional safety data through the flow/supply chain
- Enable interoperability, traceability and automation
Mission and the FS standardization Landscape

(*) Once completed and published, the Accellera FS standard is planned to be contributed to IEEE as per traditional collaboration between Accellera and IEEE
FS data = set of data needed to perform safety activities and to generate work products
Key Objectives

• Harmonize best practices and methodologies across the industry via common language

• Enable efficient interchange of data representing functional safety concepts
  • across the diverse lifecycle development tool chain and
  • among organizations engaged in distributed development

• Be comprehensive, flexible, and scalable to minimize future perceived needs for local or proprietary customization

The data model is in addition to the existing design standards
Data Model Development
Approach to Data Model Development

1. Functional Safety Analysis Process Formalization
2. Conceptual Data Model
3. Functional Safety Language

The actual exchange of information will happen through the FS Language

(*) Language Reference Manual
The conceptual data model approach

Goals:
▪ Define FS data
▪ Not to provide a reference implementation
▪ Systematic approach to define a language/format

Conceptual Data Model:
▪ Defines **WHAT** the system contains
▪ Does **NOT** define **HOW** the system should be implemented
Using the Entity Relationship model

The 3 basic tenants:

- **Entity**: The object/data describing the system to be modeled
- **Attribute**: Characteristics or properties of an entity
- **Relationship**: Dependency or association between two entities

In addition, we rely on the concept **Weak entity**, which cannot be identified by its attributes alone, but only exists in the context of another entity

FMEDA Process
Conceptual Data Model derived from the FMEDA process

Direct traceability from the data + mapping of FMEDA process to data model
Annex B: Language

Introduction

In this paper we defined a sample language for the only purpose of showing some concrete examples of usage of the Functional Safety Standard. The final LRM defined in the standard might differ from the sample used in this paper.

Following the principle of traceability, the sample language is derived directly from the conceptual data model with remarkably simple rules:

- Objects are created with “create” commands and updated with the “-update” option.
- Relationships are created with the "assign" commands.
- Weak objects are assigned a value with the "define" command.

In other words, the sample language is the implementation of the requirements defined in the conceptual data model.

A special rule stands for the Design mapping since it connects objects in the data model to objects in the design hierarchy, which are not part of the data model. The design mapping connection is described through the “-mapping” and “-exclude_mapping” options inside the design mapping relationship commands.
### Conceptual Data Model + sample commands

<table>
<thead>
<tr>
<th>FMEDA process data</th>
<th>Entity Type</th>
<th>Information Type</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMEDA</td>
<td>FMEDA</td>
<td>Object</td>
<td>create_fmeda</td>
</tr>
<tr>
<td>FS Analysis Hierarchy</td>
<td>Element</td>
<td>Object</td>
<td>create_element</td>
</tr>
<tr>
<td>FM Hierarchy</td>
<td>Failure_Mode</td>
<td>Object</td>
<td>create_failure_mode</td>
</tr>
<tr>
<td>Technology Element</td>
<td>Technology_Element</td>
<td>Object</td>
<td>create_failure_mode</td>
</tr>
<tr>
<td>Safety Mechanism Library</td>
<td>Safety_Mechanism</td>
<td>Object</td>
<td>create_failure_mode</td>
</tr>
<tr>
<td>FM Effects</td>
<td>Failure_Mode_Effect</td>
<td>Object</td>
<td>create_failure_mode</td>
</tr>
<tr>
<td>SM Mapping</td>
<td>SM-FM</td>
<td>Relationship</td>
<td>assign_SM_FM</td>
</tr>
<tr>
<td>FM Effects Mapping</td>
<td>FM-FME</td>
<td>Relationship</td>
<td>assign_FM_FME</td>
</tr>
<tr>
<td>Technology Element Mapping</td>
<td>TE-FM</td>
<td>Relationship</td>
<td>assign_TE_FM</td>
</tr>
<tr>
<td>Technology Element Mapping</td>
<td>TE-Element</td>
<td>Relationship</td>
<td>Assign_TE_Element</td>
</tr>
<tr>
<td>Design Mapping</td>
<td>Inside the TE-FM since there is no Design Hierarchy in the datamodel</td>
<td>Relationship</td>
<td>assign_TE_FM –mapping {...}–exclude_mapping</td>
</tr>
<tr>
<td>Design Mapping</td>
<td>Inside the TE-Element since there is no Design Hierarchy in the datamodel</td>
<td>Relationship</td>
<td>assign_TE_Element –mapping {...}–exclude_mapping</td>
</tr>
<tr>
<td>Calculated FR</td>
<td>FR_ISO26262</td>
<td>Weak object (*)</td>
<td>define_FR_ISO26262</td>
</tr>
<tr>
<td>Calculated metrics</td>
<td>Metrics_ISO26262</td>
<td>Weak object (*)</td>
<td>define_metric_ISO26262</td>
</tr>
<tr>
<td>Calculated FR</td>
<td>FR_IEC61508</td>
<td>Weak object (*)</td>
<td>define_FR_IEC61508</td>
</tr>
<tr>
<td>Calculated metrics</td>
<td>Metrics_IEC61508</td>
<td>Weak object (*)</td>
<td>define_metric_IEC61508</td>
</tr>
</tbody>
</table>
Use Cases

The data model implementation supports two main use cases:

1) **FMEDA evaluation**: A safety analysis is performed and described, for example, by using a command-based formalism describing the atomic actions (e.g., create the safety analysis, create a failure mode, etc.). When the user decides to generate final reports, all of the outputs are also stored in the data model. In this use case the provided authoring information is evaluated with the intent to populate the data model and to be able to generate final reports.

2) **“As is”**: A safety analysis is shared “as is,” as for example an FMEDA table or summary. In this use case there is no authoring information but only failure rates and metrics to be exchanged as outputs (for example, following a numerical evaluation of the data model) or imported as inputs.

As stated in the Accellera FS WG white paper [1], the goal for the Accellera FS standard is to work in alignment with well-established safety standards (e.g., ISO26262 [2] and IEC61508 [3]) and to facilitate their implementation. Hence, calculations and definitions are meant to be consistent with such standards (unless stated otherwise).

**Point of Discussion: Input and Output in the same format/file?**
Traceability of Data Model Development

Direct consistent mapping from process to data model objects

Direct translation from data model to language with defined rules

Traceability from:
• Requirements (FMEDA process objects and mapping) to
• Implementation of requirements (FS data model and then language commands)
Data Model White Paper

• Publish shortly after DVCon Europe 2023
• Status: Final Editorial Review
• Main Body:
  – FMEDA Process
  – Data Model
  – Associated methodology discussions
• Annexes:
  – Detailed Data Model
  – Prototype Language
  – What’s after this version
  – Repository Example
• Annexes to evolve into LRM/User Guide
# Annexes of the Data Model White Paper

<table>
<thead>
<tr>
<th>Annex</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Data model</td>
<td>34</td>
</tr>
<tr>
<td>C</td>
<td>Add-on to v0.1</td>
<td>84</td>
</tr>
<tr>
<td>B</td>
<td>Language</td>
<td>55</td>
</tr>
<tr>
<td>D</td>
<td>Repository</td>
<td>94</td>
</tr>
</tbody>
</table>

## Annex A: Data model

- Acknowledgements ................................................................. 33
- FMEDA ................................................................. 36
  - Element ................................................................. 38
  - Failure Mode ................................................................. 39
  - Technology Element ................................................................. 42
  - Safety mechanism ................................................................. 43
  - Failure mode effect ................................................................. 44
  - Mapping safety mechanisms - failure mode ................................................................. 45
  - Mapping failure mode - failure mode effect ................................................................. 46
  - Mapping technology element - failure mode ................................................................. 47
  - Mapping technology element - element ................................................................. 49
  - Define ISO/IEC 61508 failure rate ................................................................. 51
  - Define ISO/IEC 61508 metric ................................................................. 52
  - Define ISO/IEC 61508 metric ................................................................. 53
  - Define ISO/IEC 61508 metric ................................................................. 54

## Annex B: Language

- Introduction ................................................................. 55
  - Conventions ................................................................. 56
  - Safety analysis commands v0.1 ................................................................. 57
    - create_fmeda ................................................................. 58
    - create_element ................................................................. 59
    - create_te ................................................................. 60
    - create_te ................................................................. 61
    - create_attribute ................................................................. 62
    - create_collection ................................................................. 63
    - assign_attribute ................................................................. 64
    - assign_collection ................................................................. 65
    - define_fmeda ................................................................. 66
    - define_fmeda ................................................................. 67
    - define_fmeda ................................................................. 68
  - define_fmeda ................................................................. 69
  - define_fmeda ................................................................. 70

## Annex C: Add-on to v0.1

- Load Off ................................................................. 68
  - save off ................................................................. 69
  - set_scope ................................................................. 70
  - add parameter ................................................................. 71
  - attr_expr ................................................................. 72
  - assign_fmeda ................................................................. 73
  - assign_te ................................................................. 74
  - assign_te ................................................................. 75
  - assign_te ................................................................. 76
  - assign_te ................................................................. 77
  - define_fmed ................................................................. 78
  - define_fmed ................................................................. 79
  - define_fmed ................................................................. 80

## Annex D: Repository

- Example 1 ................................................................. 94
- Example 2 ................................................................. 95
- Example 3 ................................................................. 95
- Example 4 ................................................................. 95

### Step 0: Understand the difference between a language and a data model

### Step 1: Create a library of Collections of attributes

### Step 2: Create a library of Safety mechanisms

### Step 3: Create the Safety hierarchy

### Step 4: Create Failure modes and assigning Collections

### Step 5: Assign Safety mechanisms to Failure modes

### Step 6: Create Technology Elements

### Step 7: Assign Technology Elements to Failure modes - mapping

### Step 8: Create Failure mode effects and connect them to Failure modes

### Step 9: Update objects according to verification strategy

### Step 10: Create FMEDA-scope metrics

### Step 11: Create SISO-scope metrics

### Data tracing

### Equivalent tables

### Bibliography

---

[Accellera Systems Initiative](https://www.accellera.org)
Detailed Data Model
<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FMEDA Name</strong></td>
<td>String</td>
<td>Name (identifier) of the FMEDA of the project.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Enumerate (assumption-based, calculation-based)</td>
<td>Defines the source of the failure mode distribution in case a choice needs to be made. The failure mode distributions can be calculated based on: - Estimations provided with the options \texttt{fm_size} or \texttt{element_size}. - Design metrics extracted from the design mapping as specified in the \texttt{fm_mapping} and \texttt{element_mapping}. When both options (_size and _mapping) are specified for an FM, the FMEDA type will select as follows: - _size, assumption-based: The _size takes precedence over _mapping - _size, calculation-based: The _mapping takes precedence over _size</td>
<td>No</td>
</tr>
<tr>
<td><strong>ASIL</strong></td>
<td>Enumerate (None, A, B, C, D)</td>
<td>Defines the ASIL target for the FMEDA (for a given Safety Goal) according to ISO 26262. Used also to specify that the FMEDA is for ISO 26262.</td>
<td>No</td>
</tr>
<tr>
<td><strong>SIL</strong></td>
<td>Enumerate (None, 1, 2, 3, 4)</td>
<td>Defines the SIL target for the FMEDA according to IEC61508. Used also to specify that the FMEDA is for IEC61508.</td>
<td>No</td>
</tr>
<tr>
<td><strong>Analysis Type</strong></td>
<td>List of Enumerate (Permanent, Transient, All)</td>
<td>Defines the failure types to be considered and which metrics to be calculated within the safety analysis. More than one value can be specified, e.g., Analysis Type = (Permanent) or Analysis Type = (Permanent, Transient) The value ‘All’ implies all Failure Types are activated. Defined as ‘All’ instead of ‘Both’ allows for plans for more than just Transient and Permanent.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>User Defined Attribute</strong></td>
<td>List of tuples</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Point of Discussion: Required or Defaults?

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element_Name</td>
<td>String</td>
<td>Name (Identifier) of the Element.</td>
<td>Yes</td>
</tr>
<tr>
<td>Element_Description</td>
<td>String</td>
<td>Description of the intended functionality of the Element.</td>
<td>No</td>
</tr>
<tr>
<td>Element_Type</td>
<td>Enum</td>
<td>Enum {System, Element, SubElement, Component, SubComponent, Part, SubPart}</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies the type of the Element. Element_Type = Component or SubComponent</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>can only be defined if the analysis is for IEC61508, inferred from the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FMEDA entity, whether it has ASIL or SIL defined.</td>
<td></td>
</tr>
<tr>
<td>Parent_Element</td>
<td>String</td>
<td>Connects the Element to its Parent in the FS hierarchy.</td>
<td>No</td>
</tr>
<tr>
<td>FMEDA_Name</td>
<td>String</td>
<td>Connects the FS hierarchy to the FMEDA project.</td>
<td>Yes</td>
</tr>
<tr>
<td>User_DEFINEDAttribute</td>
<td>List of tuples</td>
<td>List of previously created user-defined attributes and their values.</td>
<td>No</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>SM Name</td>
<td>String</td>
<td>Name (Identifier) of the Safety Mechanism.</td>
<td>Yes</td>
</tr>
<tr>
<td>SM Description</td>
<td>String</td>
<td>Description of the SM.</td>
<td>No</td>
</tr>
<tr>
<td>FMEA Name</td>
<td>String</td>
<td>Connects the FS hierarchy to the FMEA project.</td>
<td>No</td>
</tr>
<tr>
<td>Class</td>
<td>Enumerate (HW, SW, AoU, AoU-SW, AoU-HW, user-defined)</td>
<td>Method by which the safety mechanism is to be realized.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Notes: 1) AoU is to capture when the SM is not part of the product (potentially raise a flag during FMEA integration) 2) HW allows for further specification for downstream tools</td>
<td></td>
</tr>
<tr>
<td>Class description</td>
<td>String</td>
<td>Description of the class. This is specially meant in the case in which the class is user-defined, but available for all classes.</td>
<td>No</td>
</tr>
<tr>
<td>Configurable</td>
<td>Boolean (yes, no)</td>
<td>Captures whether the SM can be turned on or off by the user/integrator. If configurable yes, then the &quot;SM-FM active&quot; attribute can be used.</td>
<td>Yes</td>
</tr>
<tr>
<td>DC Perm</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM in isolation for permanent faults.</td>
<td>Yes</td>
</tr>
<tr>
<td>DC Trans</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM in isolation for transient faults.</td>
<td>Yes</td>
</tr>
<tr>
<td>DC Lat</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM in isolation for latent faults. This attribute is only available when the ASIL target level is defined. Not available if only the SIL target is defined.</td>
<td>Yes</td>
</tr>
<tr>
<td>User Defined Attribute</td>
<td>List of tuples</td>
<td>List of previously created user-defined attributes and their values.</td>
<td>No</td>
</tr>
</tbody>
</table>

To apply a diagnostic coverage specific to an SM-FM pair, use the DC_type attribute in the SM-FM category. When SM:DC_type and SM-FM:DC_type are specified, the SM-FM:DC_type attribute takes precedence. See Mapping safety mechanism failure mode for details.
# Mapping safety mechanism - failure mode

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM Name</td>
<td>String</td>
<td>Name (identifier) of the SM applied to the FM.</td>
<td>Yes</td>
</tr>
<tr>
<td>FM Name</td>
<td>String</td>
<td>Name (identifier) of the FM covered by the SM.</td>
<td>Yes</td>
</tr>
<tr>
<td>Parent Element</td>
<td>String</td>
<td>Connects the Failure Mode to its Parent in the FS hierarchy.</td>
<td>Yes</td>
</tr>
<tr>
<td>FME DA Name</td>
<td>String</td>
<td>Connects to the FME DA project.</td>
<td>Yes</td>
</tr>
<tr>
<td>DC_Perm.Estim</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM applied to the FM for permanent faults.</td>
<td>No</td>
</tr>
<tr>
<td>DC_Trans.Estim</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM applied to the FM for transient faults.</td>
<td>No</td>
</tr>
<tr>
<td>DC_Lat.Estim</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM applied to the FM for latent faults.</td>
<td>No</td>
</tr>
<tr>
<td>DC_Perm.Meas</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM applied to the FM for permanent faults as a result of Fault Injection Activities.</td>
<td>No</td>
</tr>
<tr>
<td>DC_Trans.Meas</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM applied to the FM for transient faults as a result of Fault Injection Activities.</td>
<td>No</td>
</tr>
<tr>
<td>DC_Lat.Meas</td>
<td>Float [0, 100]</td>
<td>Diagnostic coverage of the SM applied to the FM for latent faults as a result of Fault Injection Activities.</td>
<td>No</td>
</tr>
<tr>
<td>Active</td>
<td>Boolean (yes, no)</td>
<td>Specifies whether the SM is enabled for this FM. Only accessible if the SM_Configurable attribute-yes.</td>
<td>Yes</td>
</tr>
<tr>
<td>User Defined Attribute</td>
<td>List of tuples</td>
<td>List of previously created user-defined attributes and their values.</td>
<td>No</td>
</tr>
</tbody>
</table>

DC_type value is specific to the SM-FM pair and takes precedence over the DC_type of the SM category. If such value is not specified, then the value is taken from the DC_type attribute of the SM category.
FS Standard Example
Example: Design Under Analysis
Example: FS Analysis Hierarchy

Design Under Analysis

Analysis of the Intended Functionality

FS Analysis Hierarchy

<table>
<thead>
<tr>
<th>Part</th>
<th>Subpart</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP</td>
<td>MAC</td>
</tr>
<tr>
<td></td>
<td>NON_MAC</td>
</tr>
<tr>
<td></td>
<td>MMU</td>
</tr>
<tr>
<td></td>
<td>DCFSM</td>
</tr>
<tr>
<td></td>
<td>DCTAG0</td>
</tr>
<tr>
<td></td>
<td>DCRAM</td>
</tr>
</tbody>
</table>
Example: FS Analysis Hierarchy

Create_fmeda MY_FMEDA
Create_element – type part TOP
Create_element – type subpart MAC – parent TOP -fmeda MY_FMEDA
Create_element – type subpart NON_MAC – parent TOP -fmeda MY_FMEDA
Create_element – type subpart MMU – parent TOP -fmeda MY_FMEDA
Create_element – type subpart DCFSM – parent TOP -fmeda MY_FMEDA
Create_element – type subpart DCTAG0 – parent TOP -fmeda MY_FMEDA
Create_element – type subpart DCRAM – parent TOP -fmeda MY_FMEDA
Example: FM Hierarchy

Create_fmeda MY_FMEDA
Create_element –type part TOP
Create_element –type subpart MAC –parent TOP -fmeda MY_FMEDA
Create_element –type subpart NON_MAC –parent TOP -fmeda MY_FMEDA
Create_element –type subpart MMU –parent TOP -fmeda MY_FMEDA
Create_element –type subpart DCFSM –parent TOP -fmeda MY_FMEDA
Create_element –type subpart DCTAG0 –parent TOP -fmeda MY_FMEDA
Create_element –type subpart DCRAM –parent TOP -fmeda MY_FMEDA

Create_fm FM1_DCRAM –type Mission –parent TOP.DCRAM –fmeda MY_FMEDA
Create_fm FM2_DCRAM –type Mission –parent TOP.DCRAM –fmeda MY_FMEDA
Create_fm FM_DCTAG –type Mission –parent TOP.DCTAG0 –fmeda MY_FMEDA
Create_fm FM_DCFSM –type Mission –parent TOP.DCFSM –fmeda MY_FMEDA
Create_fm FM1_MMU –type Mission –parent TOP.MMU –fmeda MY_FMEDA
Create_fm FM2_MMU –type Mission –parent TOP.MMU –fmeda MY_FMEDA
Create_fm FM_NON_MAC –type Mission –parent TOP.NON_MAC –fmeda MY_FMEDA
Create_fm FM_MAC –type Mission –parent TOP.MAC –fmeda MY_FMEDA

FS Analysis Hierarchy + FM Hierarchy

<table>
<thead>
<tr>
<th>Part</th>
<th>Subpart</th>
<th>Failure Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP</td>
<td>MAC</td>
<td>FM_MAC</td>
</tr>
<tr>
<td></td>
<td>NON_MAC</td>
<td>FM_NON_MAC</td>
</tr>
<tr>
<td></td>
<td>MMU</td>
<td>FM1_MMU</td>
</tr>
<tr>
<td></td>
<td>MMU</td>
<td>FM2_MMU</td>
</tr>
<tr>
<td></td>
<td>DCFSM</td>
<td>FM_DCFSM</td>
</tr>
<tr>
<td></td>
<td>DCTAG0</td>
<td>FM_DCTAG</td>
</tr>
<tr>
<td></td>
<td>DCRAM</td>
<td>FM1_DCRAM</td>
</tr>
<tr>
<td></td>
<td>DCRAM</td>
<td>FM2_DCRAM</td>
</tr>
</tbody>
</table>
### Example: TE and assignment

Create TE Digital_5n –type Digital –fr 1e-6
Create TE RAM –type RAM –fr 1e-5

<table>
<thead>
<tr>
<th>Part</th>
<th>Subpart</th>
<th>Failure Mode</th>
<th>Technology</th>
<th>FM_size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP</td>
<td>MAC</td>
<td>FM_MAC</td>
<td>Digital_5n</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>NON_MAC</td>
<td>FM_NON_MAC</td>
<td>Digital_5n</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>MMU</td>
<td>FM1_MMU</td>
<td>RAM</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FM2_MMU</td>
<td>Digital_5n, RAM</td>
<td>5, 25</td>
</tr>
<tr>
<td>DCFSM</td>
<td>FM_DCFSM</td>
<td>FM_DCFSM</td>
<td>Digital_5n</td>
<td>…</td>
</tr>
<tr>
<td>DCTAG0</td>
<td>FM_DCTAG</td>
<td>FM_DCTAG</td>
<td>Digital_5n</td>
<td>…</td>
</tr>
<tr>
<td>DCRAM</td>
<td>FM1_DCRAM</td>
<td>FM1_DCRAM</td>
<td>RAM</td>
<td>…</td>
</tr>
<tr>
<td></td>
<td>FM2_DCRAM</td>
<td>FM2_DCRAM</td>
<td>RAM</td>
<td>…</td>
</tr>
</tbody>
</table>

Assign TE_fm -te_name Digital_5n -fm_name FM_MAC -parent TOP.MAC -fmeda MY_FMEDA -fm_size 10

Assign TE_fm -te_name Digital_5n -fm_name FM_NON_MAC -parent TOP.NON_MAC -fmeda MY_FMEDA -fm_size 15

Assign TE_fm -te_name RAM -fm_name FM1_MMU -parent TOP.MMU -fmeda MY_FMEDA -fm_size 35

Assign TE_fm -te_name Digital_5n -fm_name FM2_MMU -parent TOP.MMU -fmeda MY_FMEDA -fm_size 5

Assign TE_fm -te_name RAM -fm_name FM2_MMU -parent TOP.MMU -fmeda MY_FMEDA -fm_size 25

....
## Example: Subpart Design Mapping

### Part/Subpart Failure Mode Technology

<table>
<thead>
<tr>
<th>Part</th>
<th>Subpart</th>
<th>Failure Mode</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP</td>
<td>MAC</td>
<td>FM_MAC</td>
<td>Digital_5n</td>
</tr>
<tr>
<td></td>
<td>NON_MAC</td>
<td>FM_NON_MAC</td>
<td>Digital_5n</td>
</tr>
<tr>
<td></td>
<td>MMU</td>
<td>FM1_MMU</td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FM2_MMU</td>
<td>Digital_5n, RAM</td>
</tr>
<tr>
<td></td>
<td>DCFSM</td>
<td>FM_DCFSM</td>
<td>Digital_5n</td>
</tr>
<tr>
<td></td>
<td>DCTAG0</td>
<td>FM_DCTAG</td>
<td>Digital_5n</td>
</tr>
<tr>
<td></td>
<td>DCRAM</td>
<td>FM1_DCRAM</td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FM2_DCRAM</td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td>DCRAM</td>
<td>FM1_DCRAM</td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FM2_DCRAM</td>
<td>RAM</td>
</tr>
</tbody>
</table>

### Diagram

```
Top
  -- dc_top
  -- Cpu
  -- MMU
     -- DCRAM
     -- dc_tag
     -- DCFSM
        -- DCTAG0
  -- MAC

Top
  -- dc_top
  -- Cpu
  -- Except
     -- Genpc
  -- Ctrl
     -- Mult_mac
  -- Dmmu_top
     -- Dmu_tlb
  -- Immu_top
     -- Ilte_tr_tram
```

### Technology Options

- Digital_5n
- RAM
- RAM

### Failure Mode Options

- FM_MAC
- FM_NON_MAC
- FM1_MMU
- FM2_MMU
- FM_DCFSM
- FM_DCTAG
- FM1_DCRAM
- FM2_DCRAM
- FM1_DCRAM
- FM2_DCRAM

---

**Note:** The image contains a table and a diagram illustrating subpart design mapping and failure modes, along with technology options. The table lists parts, subparts, failure modes, and technologies, while the diagram visualizes the hierarchical design mapping structure.
Example: Design Mapping

<table>
<thead>
<tr>
<th>Part</th>
<th>Subpart</th>
<th>Failure Mode</th>
<th>Technology</th>
<th>FM_Design_Mapping</th>
<th>FM_Design_Exclude</th>
<th>FM_Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP</td>
<td>MAC</td>
<td>FM_MAC</td>
<td>Digital_5n</td>
<td>Top.Cpu.Mult_mac</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MMU</td>
<td>FM1_MMU</td>
<td>RAM</td>
<td>{Top.Immu_top.Imu_tlb.Iltb_mr_ram, Top.Immu_top.Imu_tlb.Iltb_tr_ram}</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FM2_MMU</td>
<td>Digital_5n, RAM</td>
<td>{Top.Dmmu_top, Top.Immu_top}</td>
<td>{Top.Immu_top.Imu_tlb.Iltb_mr_ram, To...}</td>
<td>5, 25</td>
</tr>
<tr>
<td></td>
<td>DCFSM</td>
<td>FM_DCFSM</td>
<td>Digital_5n</td>
<td>Top.dc_top.dc_fsm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DCTAG0</td>
<td>FM_DCTAG</td>
<td>Digital_5n</td>
<td>Top.dc_top.dc_tag.DCTAG0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DCRAM</td>
<td>FM1_DCRAM</td>
<td>RAM</td>
<td>Top.dc_top.dc_ram.dc_ram</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FM2_DCRAM</td>
<td>RAM</td>
<td>Top.dc_top.dc_ram.dc_ram</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **FM_MAC**
- **FM_MOM_MAC**
- **FM1_MMU**
- **FM2_MMU**
- **FM_DCFSM**
- **FM_DCTAG**
- **FM1_DCRAM**
- **FM2_DCRAM**
- **MAC**
- **NON_MAC**
- **TOP**
- **DCRAM**
- **DCFSM**
- **Cpu**
Create_fmeda MY_FMEDA – type assumption_based

Create_element – type part TOP
Create_element – type subpart MAC – parent TOP - fmeda MY_FMEDA
Create_element – type subpart NON_MAC – parent TOP - fmeda MY_FMEDA
Create_element – type subpart MMU – parent TOP - fmeda MY_FMEDA
Create_element – type subpart DCFSM – parent TOP - fmeda MY_FMEDA
Create_element – type subpart DCTAG0 – parent TOP - fmeda MY_FMEDA
Create_element – type subpart DCRAM – parent TOP - fmeda MY_FMEDA

Create_fm FM1_DCRAM – type Mission – parent TOP.DCRAM – fmeda MY_FMEDA
Create_fm FM2_DCRAM – type Mission – parent TOP.DCRAM – fmeda MY_FMEDA
Create_fm FM_DCTAG – type Mission – parent TOP.DCTAG0 – fmeda MY_FMEDA
Create_fm FM_DCFSM – type Mission – parent TOP.DCFSM – fmeda MY_FMEDA
Create_fm FM1_MMU – type Mission – parent TOP.MMU – fmeda MY_FMEDA
Create_fm FM2_MMU – type Mission – parent TOP.MMU – fmeda MY_FMEDA
Create_fm FM_NON_MAC – type Mission – parent TOP.NON_MAC – fmeda MY_FMEDA
Create_fm FM_MAC – type Mission – parent TOP.MAC – fmeda MY_FMEDA

Create_TE Digital_5n – type Digital – fr 1e-6
Create_TE RAM – type RAM – fr 1e-5

Assign_TE_fm -te_name Digital_5n -fm_name FM_MAC -parent TOP.MAC -fmeda MY_FMEDA -fm_size 10 –FM_mapping Top.Cpu.Mult_mac
Assign_TE_fm -te_name Digital_5n -fm_name FM2_MMU -parent TOP.MMU -fmeda MY_FMEDA -fm_size 5 –FM_mapping Top.Dmmu_top

....
What’s Next
What’s next

• **Data Model White Paper: coming out shortly. Stay tuned!**

• LRM/User Guide + Validation
  - **Content**
    - Intent: data model content // how is info stored/exchange
    - API: interaction with data model // how to interact with info
  - **Language**
    - Formal or pseudo (even specific implementations e.g. Tcl)
    - Single or different for use cases (authoring and exchange)
    - Usage of default value

• Baseline and extensions:
  - Version 0.1: FMEDA, semiconductors
  - Post version 0.1: language feature, hierarchical support
  - Version 0.2: Safety Goals, Extension for verification support, FMEA?, system-level?
Annex C: Add-on to v0.1

This chapter describes commands that were considered by the working group, but no decision was agreed on whether accept or decline them. This chapter is for informative purposes only.

The full list of commands defined according to this extension is as follows:

- load_slf
- save_slf
- set_scope
- add_parameter
- attr_expr
- assign_fmeda_fmeda
- assign_fmeda_element

Point of discussion: data model vs integration/compression of FMEDA
Thank you

More information on the Functional Safety WG:
https://www.accellera.org/activities/working-groups/functional-safety


Data model white paper: Coming soon!!!