# EDS for complex hardware units, what is needed and how do we generate them efficiently?

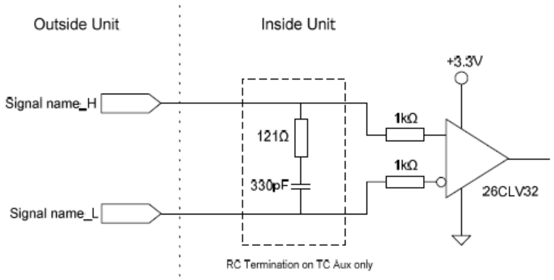
Torbjörn Hult

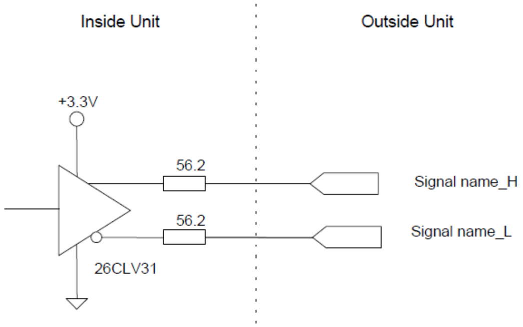
18 October 2018

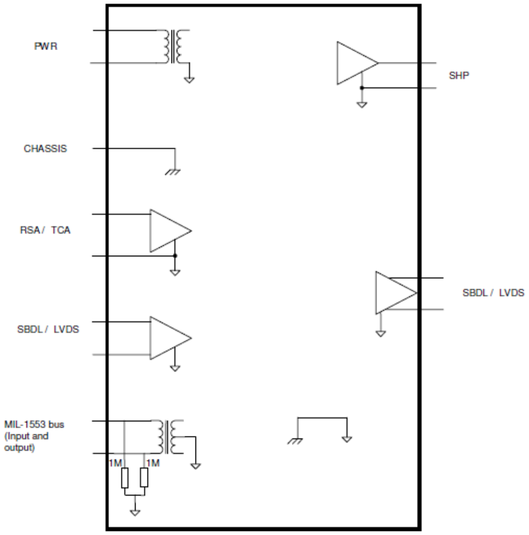
Together ahead. RUAG

# Today's ICDs for complex units

- The On-Board Computer (OBC) and the Remote Terminal Unit (RTU) are two of the most complex units in a spacecraft platform.
- The ICDs for these units are today up to180 pages, typically split in:
  - 60/100 (OBC/RTU) pages of connector information like type, pinout and electrical I/F
  - 10/45 pages of interface drawings
  - 1/1 page of unit grounding diagram
  - 1/1 page of unit block diagram
  - 50/10 pages of functional interfaces, many data structures are not PUS packets but software structures available in the Boot S/W. 20 of these pages are just to describe the configuration of the FDIR mechanisms, i.e. the OBC Reconfiguration Module.
  - 10/10 pages of TM/TC lists







# How is the “paper version” ICD generated?

- Parts of the ICDs are generated manually based on:
  - An existing ICD template for the product type or an existing ICD for the latest most similar product of the same type.
  - Unit specific configurations taken from internal and customer specifications.
  - Unit circuit diagrams.
  - Boot software ICD.

- Connector lists are generated automatically from the design data base into a Word document

# OBC HW/SW ICD

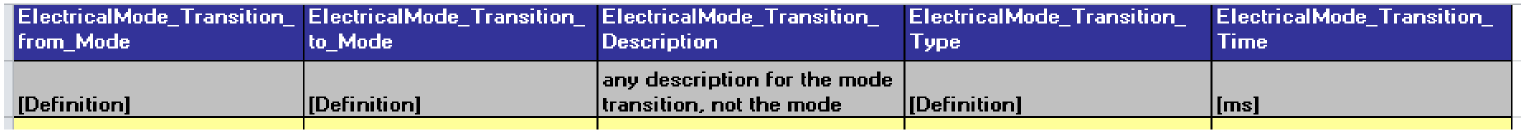
- Used internally by the OBC supplier for development of Boot S/W and Hardware Driver S/W
  - Automatically generated from the ASIC design data base
- Used by the OBC simulator developer
  - Manually generated pdf document
  - Includes the subset of the OBC registers used by the Boot and the HDSW
  - Providing all registers to the simulator developer would result in a very complex and expensive simulator
- HDSW ICD towards the Central Software is automatically generated and provided also as high level language structures in source code.
  - Can be difficult to understand for the typical system engineer

# EDS already being used for the RTU

- Generated in Sentinel-3, ExoMars TGO and MetOp SG (Excel or XML)
- Contains about 25 sheets
- Largest sheets:
  - Pin allocation, 3000 entries
  - Command message format, 600 entries
  - Acquisition message data, 400 entries
- No common EDS structure between the three programmes
  - Sentinel-3 and ExoMars TGO have the same basic I/O system but the EDS formats are different
- The RTU User Manual is still needed to understand the data in the EDS
- The RTU sometimes moves/converts data on one link to data on another link, e.g. UART data are converted to 1553 data. This can be difficult to model.

# Problems with current EDS process

- No standard EDS format
  - Too time consuming to develop automatic EDS generation → human errors still possible
  - Difficult to reuse manually generated EDS data between projects or customers
  - Difficult to predict who makes most manual errors, equipment supplier or prime
- How do we enter the required drawings?
  - As netlists?
    - Which netlist format?  PSPICE?
  - As JPEG images?
- How do we model complex behaviour like state machines with conditions?
  - Today we have something like:



| ElectricalMode_Transition_from_Mode | ElectricalMode_Transition_to_Mode | ElectricalMode_Transition_Description | ElectricalMode_Transition_Type | ElectricalMode_Transition_Time |
|---|---|---|---|---|
| [Definition] | [Definition] | any description for the mode transition, not the mode | [Definition] | [ms] |
| | | | | |

# Desired state

- An agreed EDS format is widely spread in all projects
  - Makes it possible to develop tools for automatic EDS generation
- A simple and understandable format for modelling state machines is agreed
  - Makes it possible to manually convert state diagrams to text
  - Allows for future developments of tools for automatic EDS generation from state machine definitions
- Interface drawings are not needed
- Grounding diagrams can be replaced by a parameter for each interface type

- Questions ?