SAVOIR WG
Handbook for auto-coding of HDLs for Space Applications

Alberto Urbon Aguado
Telespazio for ESA

22/10/2024

Agenda

1. Background
2. Challenges
3. Use cases
4. Goals and Scope
5. Potential Contents
6. Timeline
7. Q&A

1. Background

Digital electronics development = **Models** = abstractions of reality

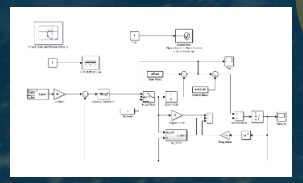
Simplified picture of where we are, and steps involved…

High Level

```
int array[size], key,i;

// Taking Input In Array
   for(int j=0;j<size;j++){
   cout<<"Enter "<<j<<" Element: ";
   cin>>array[j];
   }

//Your Entered Array Is
   for(int a=0;a<size;a++){
      cout<<"array[ "<<a<<" ]  =  ";
      cout<<array[a]<<endl;
   }
```



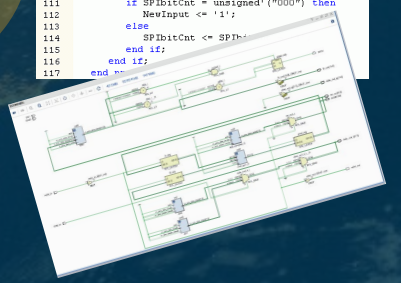
High Level Synthesis

Auto-coding

Traditional Input Level

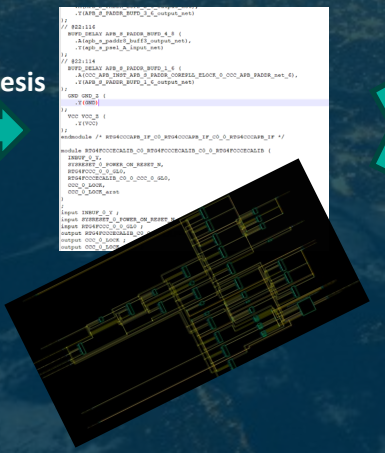
RTL HDL

```
97  ----------------------------------------------------------------------
98  -- Process Name:
99  -- Revisions:
100 -- Revision
101 -- Additional Comments:
102 --
103 ----------------------------------------------------------------------
104     SPIlatchseq : process(Sclk, CS, SDI)
105     begin
106         if CS /= CPLD_SELECT then
107             SPIbitCnt <= "111";
108             NewInput <= '0';
109         elsif Sclk = '0' and Sclk'event then
110             InputReg <= InputReg(6 downto 0) & SDI;
111             if SPIbitCnt = unsigned'("000") then
112                 NewInput <= '1';
113             else
114                 SPIbitCnt <= SPI
115             end if;
116         end if;
117     end
```

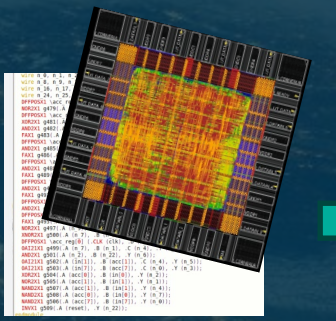


Synthesis

PostSyn Netlist



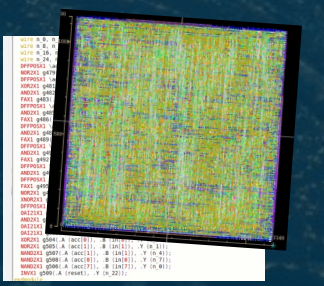
Place & Route



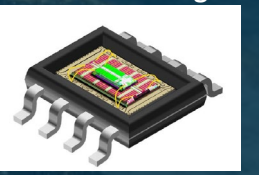
FPGA programmin

Floorplan, Place, CTS & Route

P&R design - PostLayout netlist



Digital ASIC manufacturing



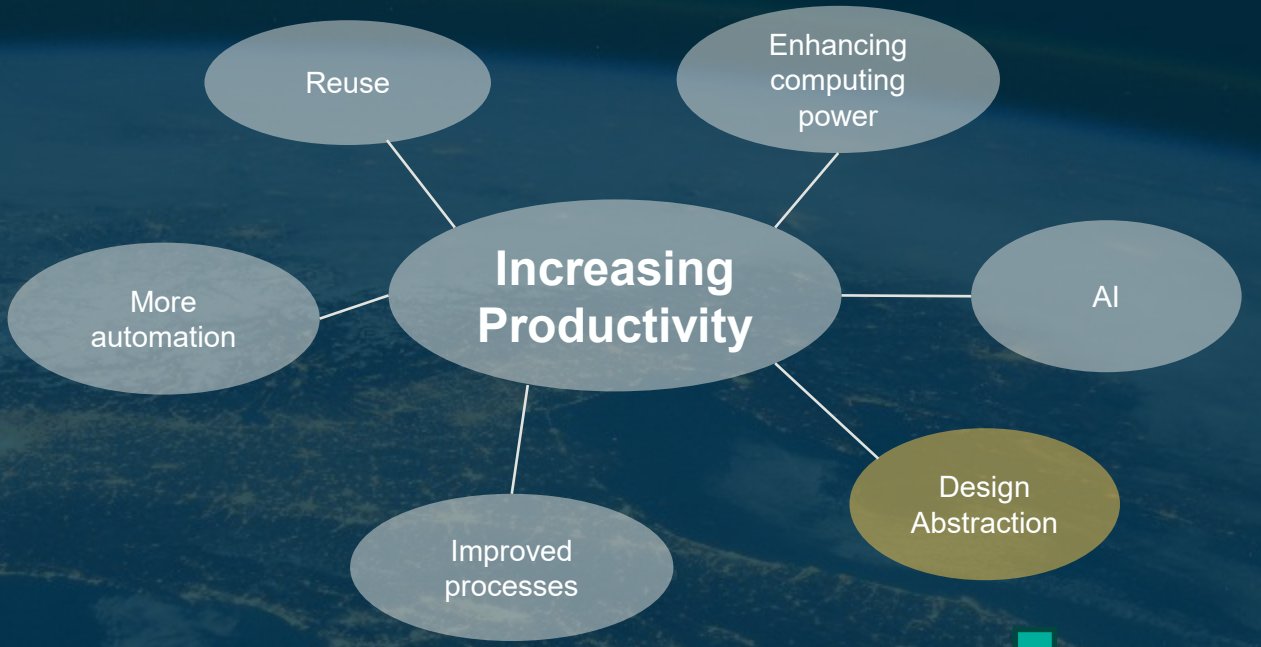
Current ECSS standard coverage (ECSS-E-ST-20-40C)

2. Challenges

Today´s challenges in ad-hoc HW devs...

- Higher complexity
- Shorter times to market
- Increasing development efforts

..require to boost productivity..



High Level Models, which can be *language based* or *graphical/mathematical based* are widely used for digital electronics design in certain space applications as:

- Digital Signal Processing
- Image and Video Processing
- Cryptography

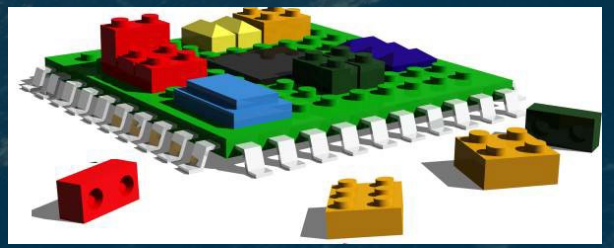
Most used tools in our domain:

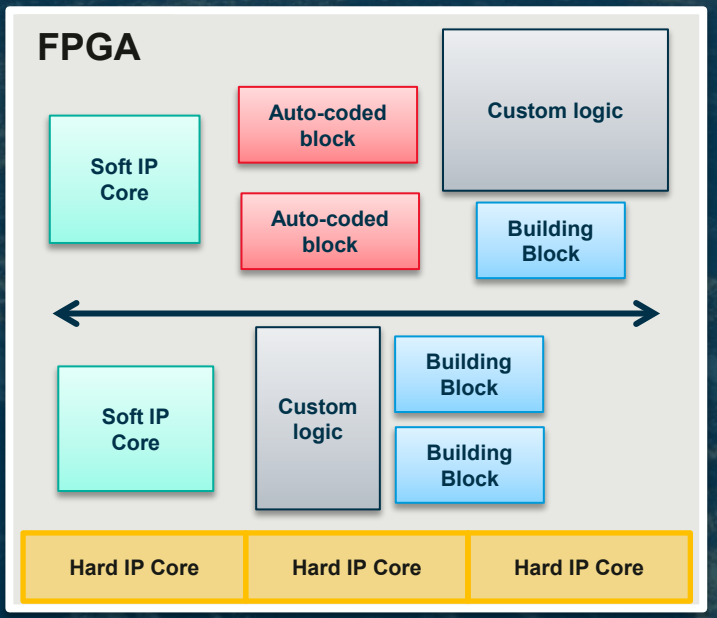
- Matlab/Simulink HDL Coder
- PoliMi Panda/Bambu
- AMD/Xilinx Vitis HLS
- Microchip Smart HLS
- Siemens Catapult HLS

3. Use cases

- How is HDL auto-coding used these days?
  - To tackle specific data processing modules where low level modelling is challenging (complex arithmetic, many data types conversions, multi domain interactions...)

- FPGA development available tools:
  - Hard IP instantiation
  - Soft IP instantiation
  - Building Block reuse and tweak
  - Custom logic development
  - High Level Modelling + Auto-coding





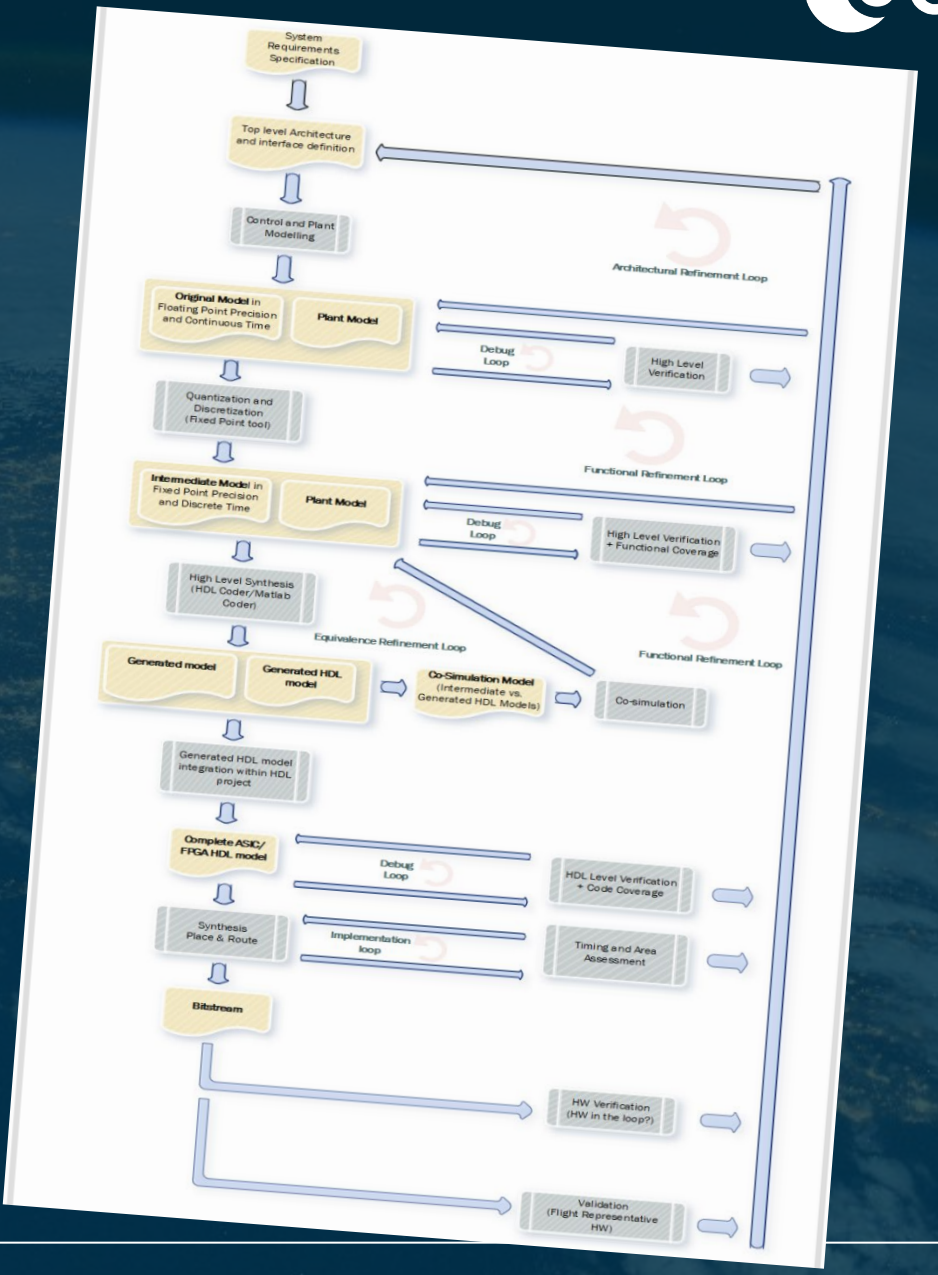
4. Goals and Scope

1. To produce a set of auto-coding guidelines at three levels: *Mandatory, Strongly Recommended, Recommended.* Not to be defined with requirements like mindset, but to be used as desired.
2. To capture generic aspects in the main part of the HB, no matter the *High-Level Model* flavour, nor the auto-coding tool
3. To capture MT/SL+HDL Coder specific guidelines within an Annex, due to its widespread use in our industry
4. To reduce issues found during developments using this philosophy
5. To propose potential traceability and means of compliance towards ASIC/FPGA/IP development standard (ECSS-E-ST-20-40C), since the paradigm is not covered by any standard
6. To agree on how to produce HDLs reliably from HL models
7. To define design flows including High Level Verification
8. To agree on the most interesting Verification approaches
9. To agree on HL Model reuse, reports generation, mitigation techniques at HL model level…
10. To increase awareness and knowledge of these technologies and to ease its deployment in the EU space ecosystem
11. And last but by far not least, to gather a group of experts on the topic to discuss approaches, pitfalls, upsides and downsides…

Its scope would be the generation of technology independent VHDL code for Space Segment applications

5. Potential contents

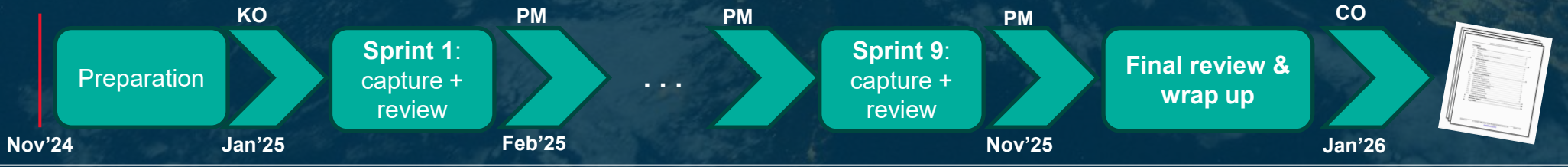
- Glossary and definitions
- Traditional HDL design capture flow vs HL design capture flow → draft..
  →
- Engineering roles involved
- Intermediate models: from original HL to HDL
- Interfaces and hierarchy (top level, register boundaries..)
- Data types, fixed point, floating point, quantization, rounding, saturations…
- Clocking and reset, when clocks and resets don't exist
- Memory allocation and management, DSPs, control structures as FSMs…
- Radiation Mitigation techniques in high level models (TMRs, ECCs, FSMs…)
- Verification approaches as HL functional verification, HL functional coverage, FPGA in the Loop, HW in the Loop, HL Model vs HDL Co-simulation, Test vectors, Formal equivalence of HL model vs HDL…
- Optimizations: delay balancing, loop unrolling, loop pipelining, pipelining (control vs data, no feedback..)
- Potential traceability towards ECSS-E-ST-20-40C
- Reporting and reusing



6. Timeline

Plan!

- **Preparation**: Nov/Dev'24
  - Skeleton preparation
  - WG members nomination + confirmation of availability
  - Collaborative environment for remote working
  - ECSS-E-ST-20-40C and other relevant docs study by WG members
- **KO**: Jan'25
  - WG members intro and related background
  - HB presentation: goals + sections + collaborative work environment
  - Skeleton proposal sharing
  - Work breakdown and distribution of tasks among members
- **Production**: Jan'25 to Nov'25
  - 4 weeks sprints including review of other's inputs before PM
  - Monthly PM on progress and to discuss raised discrepancies
  - Duration: 10 months of work (in reality, 9, due to summer break)
  - Last month to wrap up and polish format
- **Wrap up and CO**: Jan'26



7. Q&A

Any Questions!??

Thank you!