# What's next? Beyond HDLs in space FPGA developments

To be presented at ADCSS'23

Alberto Urbón Aguado – ESA TEC-EDM on behalf of Telespazio
14/11/2023

# Outline

1. Intro: levels of abstraction in digital electronics design
2. Today's challenges on space FPGAs
3. Increasing productivity
4. High Level Synthesis
5. Model Based Design
6. Follow-on: potential ECSS Handbook on auto-coding for Space FPGAs
7. References
8. Q&A

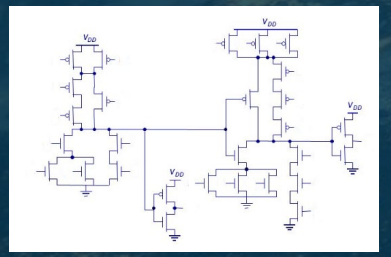
# 1. Intro: levels of abstraction in digital electronics design

From the early years in digital electronics design, engineers have embraced abstraction-based design methodologies. Even the transistor level schematics, technological details are overlooked.

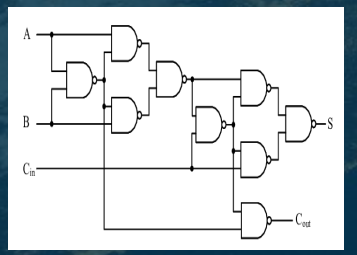
The best way to cope with larger complex circuits has always been to "zoom out", when possible

**Models at all these levels can describe the same circuit!**

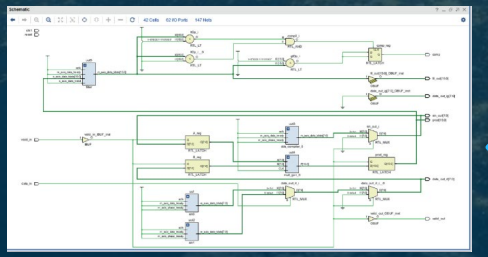
Transistor Level Schematics Model



Gate Level Schematics Model



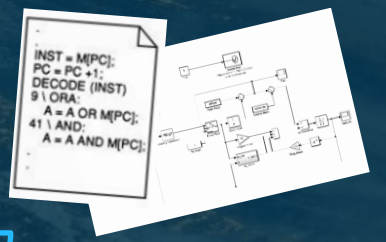

RTL Schematics Model



RTL HDL Model

```
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 <= SPIbitCnt - '1';
115             end if;
116         end if;
117     end process;
```

High Level Model




Synthesis

High Level Synthesis

1960's — 1980's — 2000's

Least abstract representation

Most abstract representation

# 2. Today's challenges: Device dimension

| | RTAX-2000 | RT Polarfire | Δ | RT Polarfire 2 |
|---|---|---|---|---|
| Release year | 2001 | 2020 | 20 years | 2024? |
| Technology node | UMC Bulk 150nm | UMC Bulk 28nm | > density, aprox. x1/6 | ? |
| CMem | Antifuse (OTP) | Flash (reprogrammable) | > versatility<br>< constrained flow | ? |
| Programmable Logic Cells | 21.5K | 481K | aprox. x22 | ? |
| Internal RAM | 288 Kbits | 33Mbits | aprox. x117 | ? |
| Instantiable FFs | 10.7K TMRed (RHBD die) | 481K (160.3K if TMRed) | aprox. x44<br>aprox. x15 if TMRed | ? |
| Max. frequency | 100MHz | 450MHz | x4.5 | ? |
| Hard IPs | None | - EDAC for RAMs<br>- PLLs<br>- System Controller<br>- HSSLs & PCIe | NA | ? |

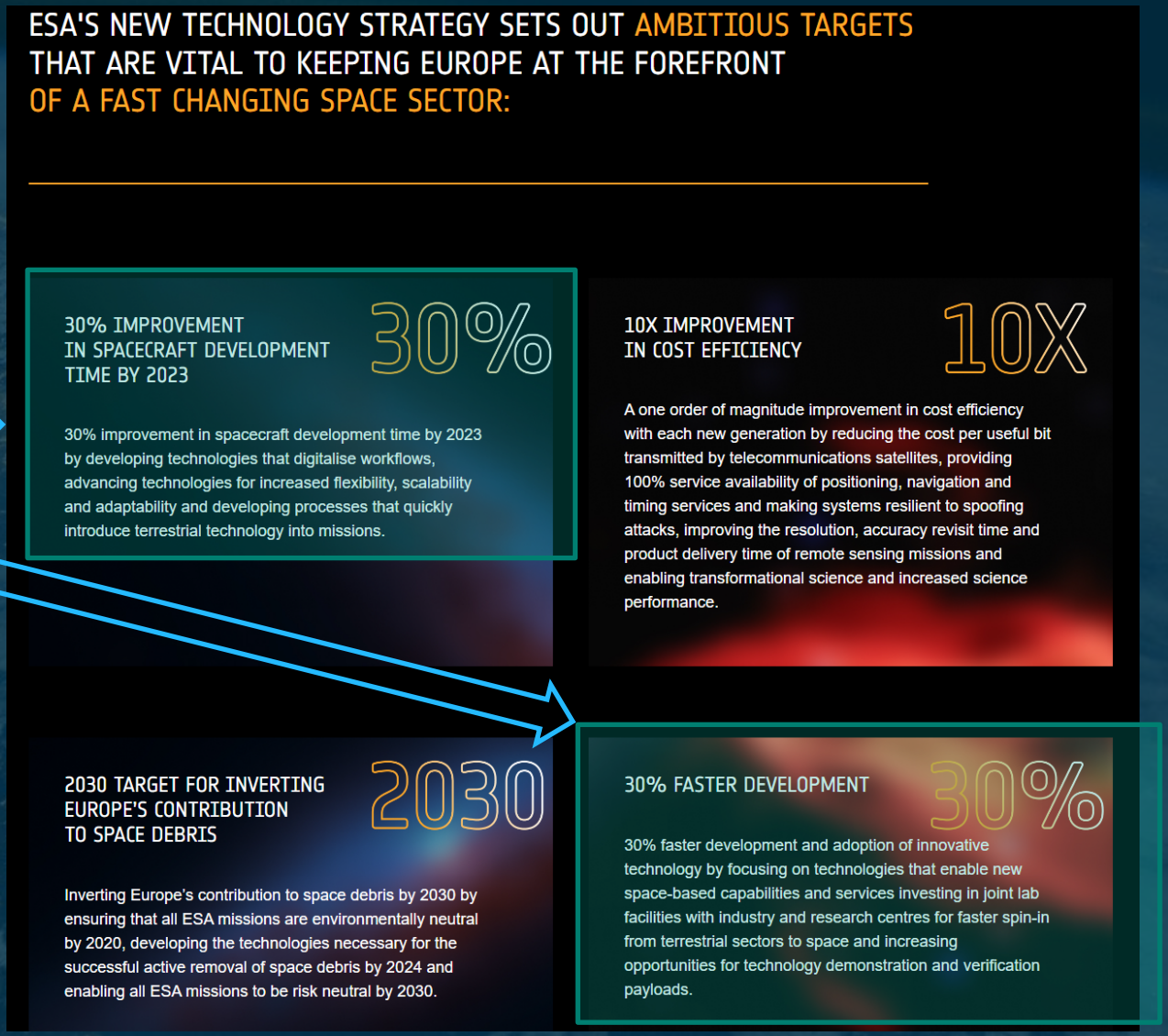
*Evolution of Microchip's flagship FPGA for space in roughly 20 years*

As number of registers increase linearly, development and verification efforts increase exponentially!

Designing in the same manner as 20 years ago makes no longer sense!
Nowadays, designs from scratch are impractical and unfeasible if we don't get abstracted!

# 2. Today's challenges: Time to market

On top of that, ESA's roadmap in technology is aiming to improve the S/C development time by a 30%...



# 2. Today's challenges: Design complexity evolution.

## Image Compression Algorithms

AIRBUS

### CORECI [7]

(Compression Recording & Ciphering)

- Year: 2006-2007
- Algorithm: MRCPB, comparable to CSSDS-122.0.B-1 (2007)
- Method: hand-coded HDL
- Implementation: WICOM ASIC in ATMEL MH1RT @ 350 nm (CWICOM ASIC in ATMEL ATC180RHA @ 180nm for CSSDS 122.0.B)
- Performance: 2D 13b images at 20/25 Mpix/s
- Missions: Pleiades, x6 ASICs + x2 FPGA in Sentinel 2, Spot 6/7, Solar Orbiter..

Evolution

### CORECI v2 [8]

- Year: 2017-2019
- Algorithm: GOLRA, improvement from CSSDS-122.0.B-2 and less resources
- Method: hand-coded HDL
- Implementation: RTG4 FPGA in 65nm with filling ratio >80%, very complex
- Performance: 180 Mpix/s
- Missions: Pleaides Neo, CO2M (partial)...

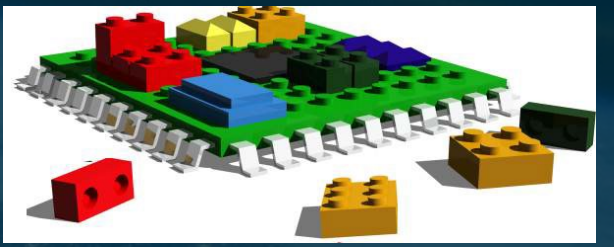
Evolution

### CORECI v3 ?

- Year: 2022..
- Algorithm: ?
- Method: ?
- Implementation: NG-Ultra RHBD 28nm
- Performance: ?
- Missions: ?

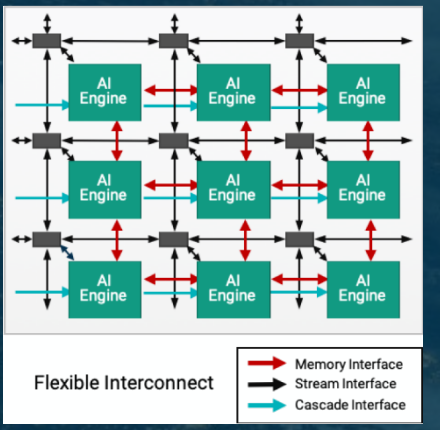
# 3. Increasing productivity

How to cope with
- bigger devices
- shorter developments
- more functionalities



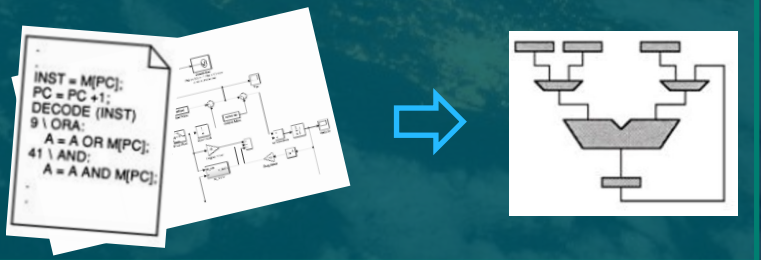
Techniques to increase productivity in uElectronics are related to design abstraction:

- **Hard IPs:** provided as instantiable logic that comes in silicon within the IC
- **Soft IPs:** instantiable fully verified and documented blocks: ESA IP Core Technical Requirements and ECSS-E-ST-20-40C – ASIC, FPGA and IP Core engineering
- **Building Blocks:** reusable modules as well, although in this case they might not be fully verified/documented, they shall be considered as if the code was newly created.
- **AI Inference:** predefined architecture of processing units that will compute the sequence of operations that have been previously defined during the training phase of the solution (e.g. AI processor, TPUs, AI engines…)




*AMD/Xilinx*

- **HLS (High Level Synthesis):** Automated design error-free process that converts an abstract high-level behavioural description/specification of a design, to an equivalent RTL model to use it as input to ASIC/FPGA implementation. Between 50%-70% reduction of the development time [2], but less optimized results in area (and frequency)



# 4. High Level Synthesis

Automated design error-free process that converts an abstract high-level behavioural description/specification of a design, to an equivalent RTL model to use it as input to ASIC/FPGA implementation. The high-level design can be expressed in two forms mainly (5 as per [2]):

- Language based: behavioural models are based in high level languages as C/C++/SystemC/Matlab, mainly
  - Vitis HLS from Xilinx (Vivado HLS, AutoPilot):
    - Source: C/C++/OpenCL
  - Synphony C Compiler from Synopsys:
    - Source: C/C++
  - Catapult HLS from Mentor/Siemens
    - Source: C/C++/SystemC
  - Smart-HLS from Microchip (formerly LegUp)
    - Source: C++
  - HLS Compiler from Intel/Altera (formerly a++)
    - Source: C++
  - Stratus HLS from Cadence Design Systems
    - Source: C/C++/SystemC
  - Panda/Bambu from Politechnico di Milano
    - Source: C/C++
  - CCubed from Univ. Western Macedonia
    - Source: C/Ada
  - BlueSpec Compiler from BlueSpec
  - Impulse-C CoDeveloper from Impulse Accelerated Technologies
  - CyberWorkBench from NEC
  - C-to-Verilog from C-to-Verilog
  - eXCite from Y Explorations
  - ParC C++ extended for parallel processing and hardware description

Some of these tools are very mature!

- Graphical and mathematical based: process that enables developments of dynamic systems by means of an abstract and virtual representation
  - Matlab/SL HDL Coder from Mathworks:
    - Source: Matlab code/SL graphical models/FSMs
  - Synphony Model Compiler ME from Synopsys (deprecated):
    - Source: Matlab code/SL graphical models/FSMs

(Highlighted the ones that are used the most in our sector)

# 4. High Level Synthesis.

## Image Compression Algorithms

### CORECI [7]

(Compression Recording & Ciphering)

- Year: 2006-2007
- Algorithm: MRCPB, comparable to CSSDS-122.0.B-1 (2007)
- Method: hand-coded HDL
- Implementation: WICOM ASIC in ATMEL MH1RT @ 350 nm (CWICOM ASIC in ATMEL ATC180RHA @ 180nm for CSSDS 122.0.B)
- Performance: 2D 13b images at 20/25 Mpix/s
- Missions: Pleiades, x6 ASICs + x2 FPGA in Sentinel 2, Spot 6/7, Solar Orbiter..

Evolution

### CORECI v2 [8]

- Year: 2017-2019
- Algorithm: GOLRA, improvement from CSSDS-122.0.B-2 and less resources
- Method: hand-coded HDL
- Implementation: RTG4 FPGA in 65nm with filling ratio >80%, very complex
- Performance: 180 Mpix/s
- Missions: Pleaides Neo, CO2M (partial)...

Evolution

### CORECI v3 ?

- Year: 2022..
- Algorithm: ?
- Method: ?
- Implementation: NG-Ultra RHBD 28nm
- Performance: ?
- Missions: ?

### HyperSpectral Loseless & Near lossless compression [1]

Alternative approach when tech. allows and tight schedules???

- Year: 2019-2022
- Algorithm: pre-study with CSSDS-123.0.B-1 + CSSDS-123.0.B-2
- Method: Catapult HLS & Xilinx Vitis HLS
- Implementation: pre-study NX NG-Medium + Xilinx KU060 FPGAs
- Mission: CHIME

| | Pre-study CSSDS-123.0.B-1 in NG-Medium | | | | | | |
|---|---|---|---|---|---|---|---|
| | CC | FF | BRAMs | DSPs | LUTs | Max. Freq. | Dev. Time |
| Vitis HLS | 4506 | 5268 | 25 | 10 | 8282 | 16.2 | 3 weeks |
| Catapult HLS | 4130 | 8478 | 0 | 1 | 15041 | 16.0 | 2 weeks |
| SHyLoC 1.0 (hand) | 2743 | 3083 | 43 | 6 | 3806 | 39.1 | 3 months |

# 5. Model Based Design

Mathematical and graphical process that enables developments of dynamic systems by means of an abstract and virtual representation (behavioural model).

MBD is not suitable for all sort of designs, but for certain applications, the approach is superior.

Normally targeting concrete processing blocks

Abstraction from details

Virtual and graphical approach

Specification on its own (dismissing paper reqs)

Easy to share/exchange

Possibility to reuse blocks/IPs

Multi-domain approach (continuous, discrete, physical, FSM, event driven.. )

Multi-disciplinary toolset (digital, analogue, power..)

Allows rapid prototyping and design space exploration at early stages

Easy portability among different technologies

Version control compatible (Git)

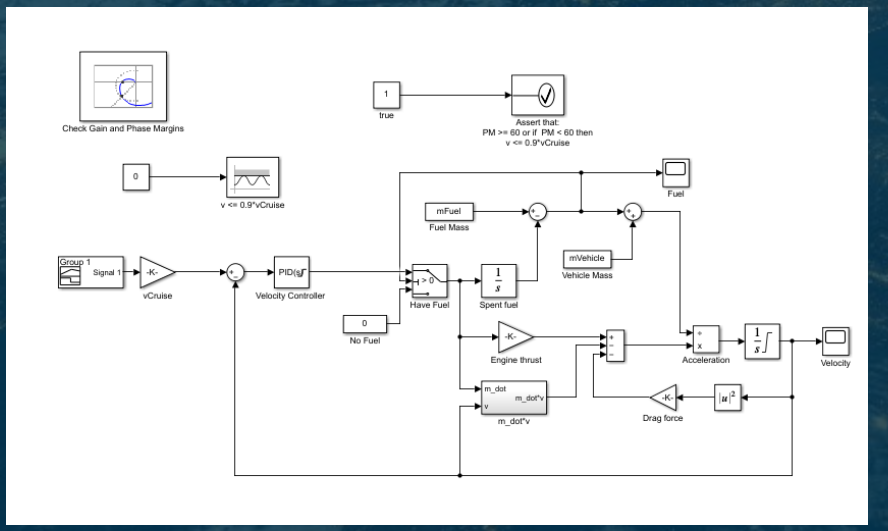
Higher level simulations (Algorithm freq. vs. Implementation freq.)

Debugging at High Level:
-High Level Model/High Level Coverage
-FPGA in the Loop
-Hardware in the Loop

Opens up uElectronics to engineers from other disciplines
Great lack of RTL engineers!

Well targeted solution for load partitioning in SoC devices!

- SW auto code generation
- HDL auto code generation
- GPU auto code generation



# 6. Follow-on: potential ECSS HB on auto-coding for Space FPGAs

*Some ESA FPGA/ASIC/IP suppliers using this approach*

**MBD HLS is well suited for ad-hoc solutions in:**

- Image and video processing
- Digital signal processing
- Robotics
- Motor control
- Digital power control
- GNC

Normally targeting concrete processing blocks

**Status**

- Proposal presented to ECSS TA in March'23 – approved, but on hold
- Focus on auto-coding guidelines: Generic and MT/SL+HDL Coder specific

**Why?**

- To reduce issues found during developments using this philosophy
- Because MT/SL is widely used in Industry/Academia/Agencies
- Paradigm not covered by any standard/HB
- To agree on how to produce HDLs automatically and reliably from HL models
- To define design flows and V&V approaches
- To propose radiation mitigation techniques at model level
- To clarify how to reuse models
- To agree on the most interesting reports
- To keep coherence and link these guidelines to the ASIC/FPGA/IP development standard (ECSS-Q-ST-60-03C and ECSS-E-ST-20-40C)
- To ease its deployment in a similar manner across the EU space ecosystem
- To trigger its use in SMEs with bigger difficulties to adopt new flows

# 7. References


- [1] Y. Barrios, R. Neris, R. Guerra, S. López and R. Sarmiento (IUMA), "Speeding up FPGA Prototyping on Space Programs with HLS Workflow. Use Case: Video Compression On-board Satellites" 2022 37th Conference on Design of Circuits and Integrated Circuits (DCIS), Pamplona, Spain, 2022, pp. 01-06, doi: 10.1109/DCIS55711.2022.9970056.
- [2] S. Lahti, P. Sjövall, J. Vanne and T. D. Hämäläinen, "Are We There Yet? A Study on the State of High-Level Synthesis" in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 5, pp. 898-911, May 2019, doi: 10.1109/TCAD.2018.2834439.
- European Data Handling & Data Processing Conference (October 2023):
  - [3] J. Moreno, R. Regada, J.M. R. Bejarano (TAS in Spain), "Applying Model-Based Design and Model-Based Systems Engineering for High-Level Design and Verification in Space Application"
  - [4] J. Valverde (Mathworks), "ECSS Compliance using Model-Based Design: A Vision-Based Navigation System on an FPGA"
  - [5] S. Lee, R. Salvador, A. Kritikakou, O. Sentieys, J. Galizzi, E. Casseau "High-Level Synthesis (HLS)-Based On-board Payload Data Processing considering the Roofline Model"
  - [6] I. Masar (TTTech), "Model-Based Design and Rapid Prototyping of Distributed Real-Time Applications based on Time-Triggered Ethernet"
- [7] H. Pelon, U. Lonsdorfer (ADS - former EADS Astrium), "A Versatile Wavelet Image Compression Module in Sentinel 2" 2008 On-Board Payload Data Compression Workshop (OBPDC)
- [8] C. Le Lann, G. Rozier (ADS), "Image Compression on Pleiades Neo" 2022 On-Board Payload Data Compression Workshop (OBPDC)

# 8. Q&A

Any Questions!??

Thank you!