How to get custom SoCs done fast and correct

Pascal Pieper *Pascal.Pieper@DLR.de*
German Aerospace Center, Bremen, Germany

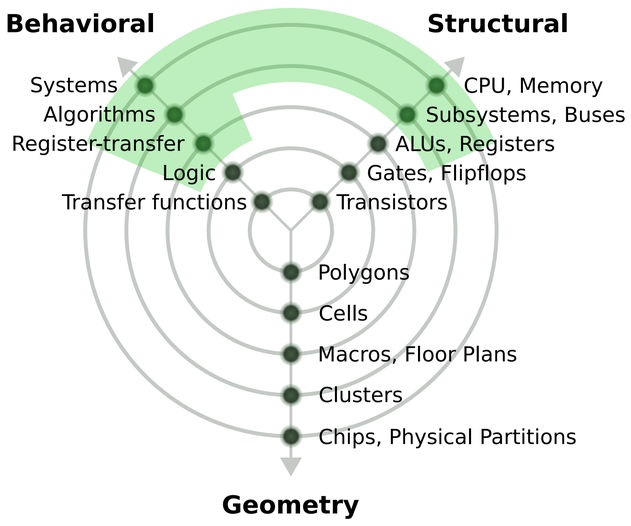
RISC-V IN SPACE

DLR Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

1. Introduction

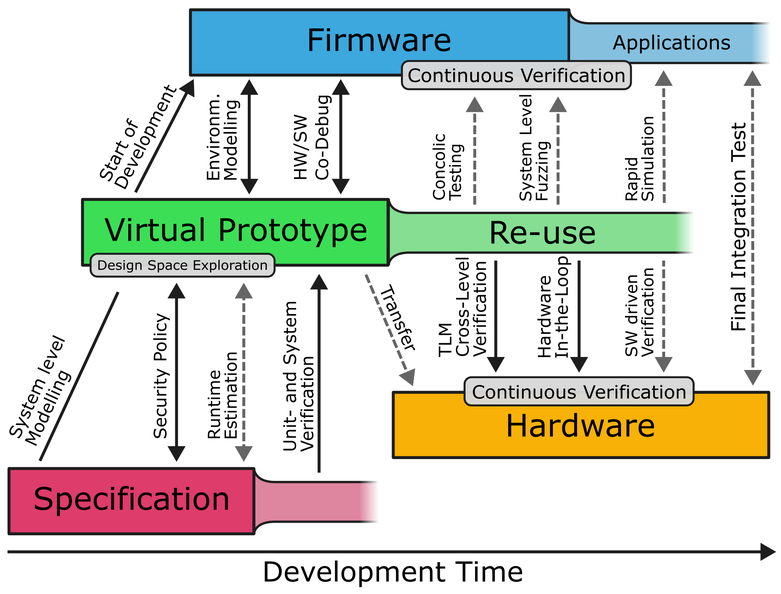
Virtual Prototypes (VPs) are incredibly useful for the complex *System on Chip* (**SoC**) design process. A VP creates the possibility to **design, evaluate and verify** an executable prototype of the system in an early design stage by modelling the future *Hardware* (HW) on an **system** / **architectural** level.

The VP-enhanced system design flow enables both the **iterative design evaluation** and **parallel development** of the (actual) HW and *Software* (SW) very early in the product conception phase. Additionally, VPs can be used as **golden reference models** with test and verification methods for comparison between the system level behaviour and the actual HW.

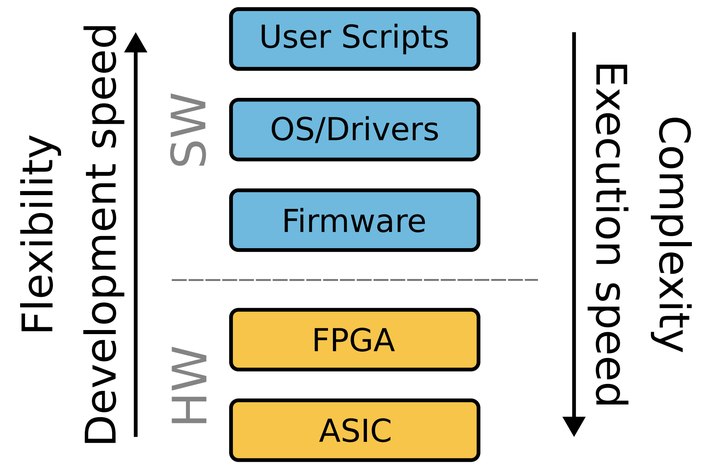


2. Process Speedup

The main goal of this poster is to encourage companies and individuals to use VPs and invest in the comparatively low effort to create a correctly configured VP to benefit from new possibilities. Most notably, to show improvements in **development speedup**, **SW quality**, **HW quality**, and **accessibility**.

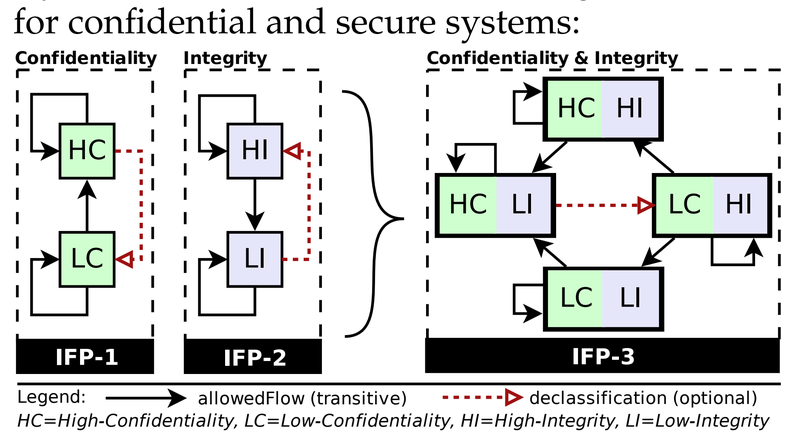


The speedup mainly relies on the **HW/SW co-development** (shift left principle) and **early design space exploration**. This allows problems like HW-SW-Partitioning [6] to be solved **before** the existing design can't be changed.

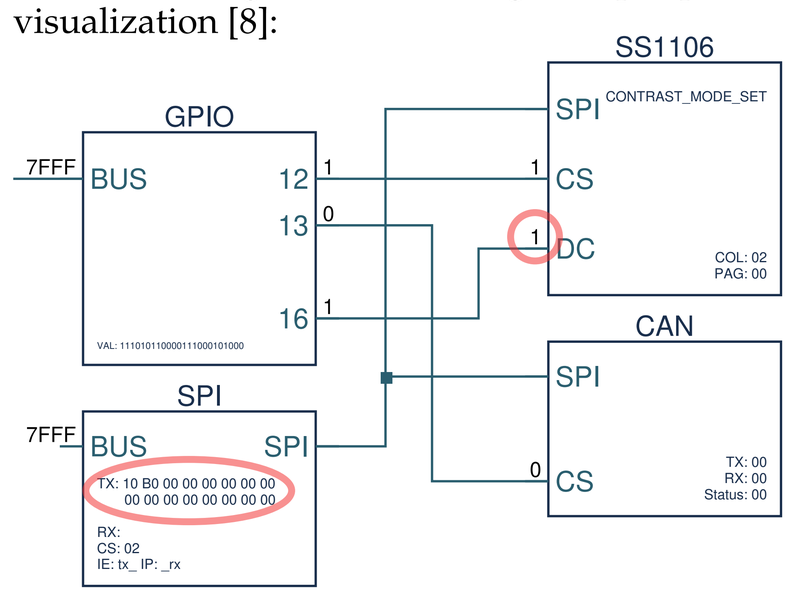


3. SW quality

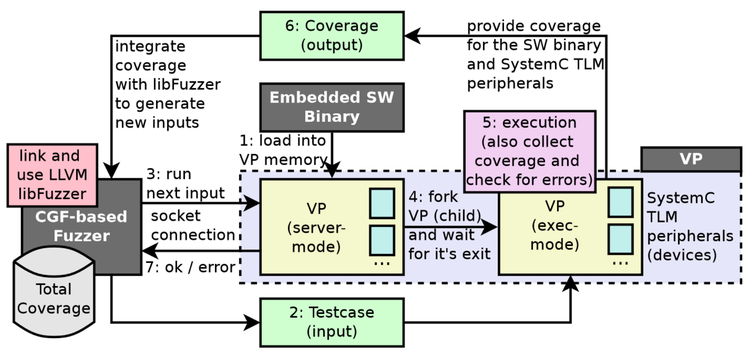
Dynamic Information Flow Tracking [7] for confidential and secure systems:



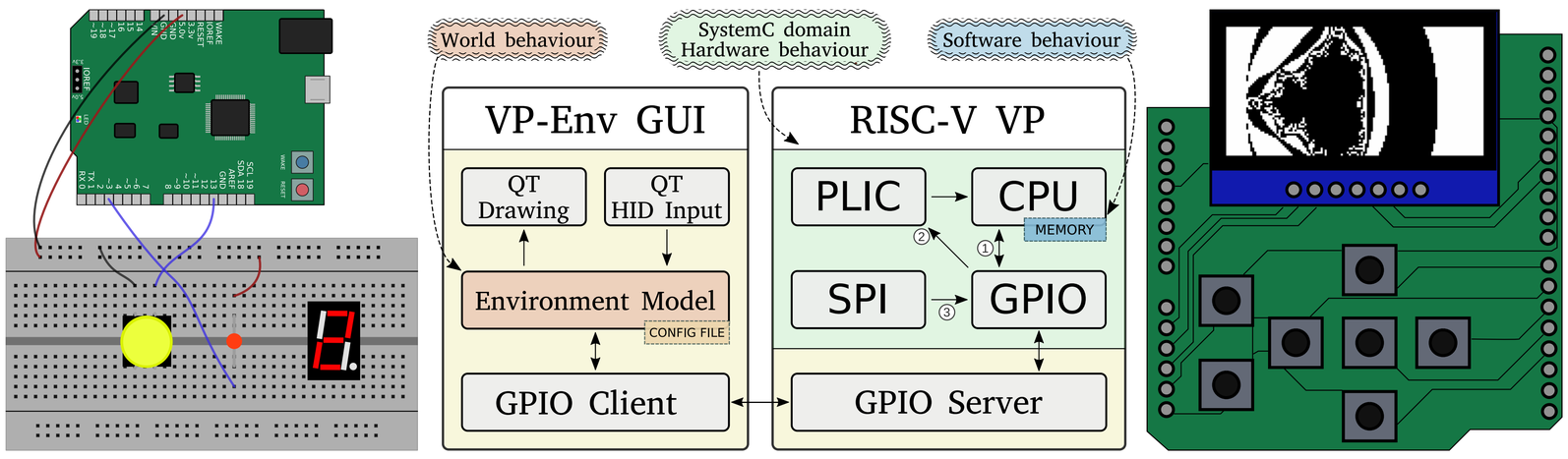
Firmware design understanding with peripheral visualization [8]:



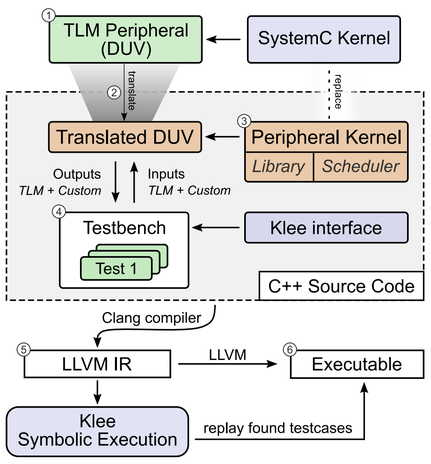
Symbolic and concolic execution of low-level SW that interacts with HW [9, 10, 11]:

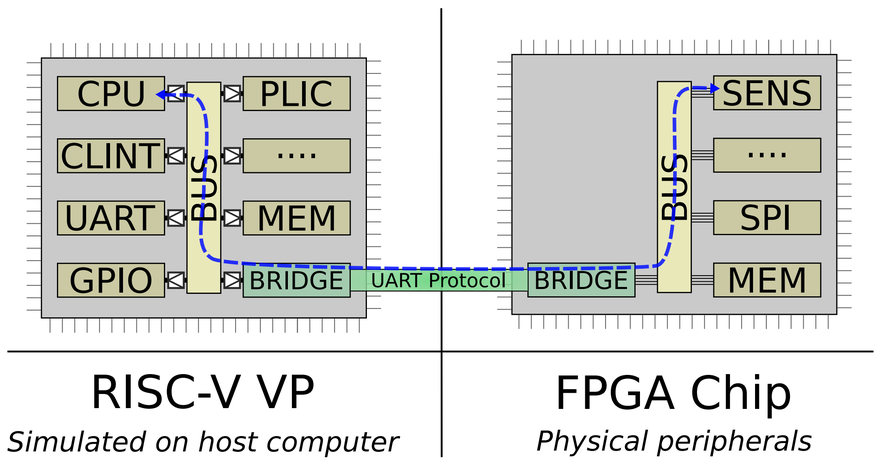


4. HW Modeling and Quality



- Interactive environment modeling [12] for off-chip devices on a PCB
- Constrained random verification [13] for HDL-vs-TLM comparison
- Symbolic execution of SystemC models [14], with HDL-synthesis methods [15]
- Hardware-in-the-Loop testing [16] or co-simulation [17] for focus on unique selling point





5. References

Sorted by execution speed vs. low-level detail: QEMU [1], RISC-V JIT [2], Renode [3] and TLM-based platforms such as the RISC-V VP [4, 5].

[1] F. Bellard, "Qemu, a fast and portable dynamic translator," in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ser. ATEC '05. USA: USENIX Association, 2005, p. 41.

[2] V. Herdt, D. Große, S. Tempel, and R. Drechsler, "Adaptive simulation with virtual prototypes in an open-source risc-v evaluation platform," *Journal of Systems Architecture*, vol. 116, p. 102135, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121001016

[3] (2022) Renode. [Online]. Available: https://renode.io/

[4] V. Herdt, D. Große, P. Pieper, and R. Drechsler, "RISC-V based virtual prototype: An extensible and configurable platform for the system-level," *JSA*, vol. 109, p. 101756, 2020.

[5] S. Ahmadi-Pour, V. Herdt, and R. Drechsler, "The microrv32 framework: An accessible and configurable open source risc-v cross-level platform for education and research," *Journal of Systems Architecture*, vol. 133, p. 102757, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762122002429

[6] F. Vahid, "What is hardware/software partitioning?" *SIGDA Newsl.*, vol. 39, no. 6, p. 1, jun 2009. [Online]. Available: https://doi.org/10.1145/1862900.1862901

[7] P. Pieper, V. Herdt, D. Große, and R. Drechsler, "Dynamic Information Flow Tracking for Embedded Binaries using SystemC-based Virtual Prototypes," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[8] P. Pieper, R. Wimmer, G. Angst, and R. Drechsler, "Minimally invasive HW/SW co-debug live visualization on architecture level," in *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, ser. GLSVLSI '21. New York, NY, USA: ACM, 2021, p. 321–326.

[9] V. Herdt, D. Große, J. Wloka, T. Güneysu, and R. Drechsler, "Verification of embedded binaries using coverage-guided fuzzing with systemc-based virtual prototypes," in *Proceedings of the 2020 on Great Lakes Symposium on VLSI*, ser. GLSVLSI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 101–106. [Online]. Available: https://doi.org/10.1145/3386263.3406899

[10] S. Tempel, V. Herdt, and R. Drechsler, "SymEx-VP: An open source virtual prototype for os-agnostic concolic testing of iot firmware," *Journal of Systems Architecture*, vol. 126, p. 102456, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762122000480

[11] ——, "Automated detection of spatial memory safety violations for constrained devices," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2022, pp. 160–165.

[12] P. Pieper, V. Herdt, and R. Drechsler, "Advanced embedded system modeling and simulation in an open source RISC-V virtual prototype," *JLPEA*, vol. 12, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2079-9268/12/4/52

[13] S. Ahmadi-Pour, V. Herdt, and R. Drechsler, "Constrained random verification for risc-v: Overview, evaluation and discussion," in *MBMV 2021; 24th Workshop*, 2021, pp. 1–8.

[14] P. Pieper, V. Herdt, and R. Drechsler, "Verifying SystemC TLM peripherals using modern C++ symbolic execution tools," in *2022 59th ACM/IEEE Design Automation Conference (DAC)*, 2022, pp. 1–6.

[15] P. Coussy, A. Takach, M. McNamara, and M. Meredith, "An introduction to the systemc synthesis subset standard," 10 2010, pp. 183–184.

[16] P. Pieper, S. Ahmadi-Pour, and R. Drechsler, "Virtual-peripheral-in-the-loop: A hardware-in-the-loop strategy to bridge the VP/RTL design-gap," in *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES+ISSS '23. New York, NY, USA: Association for Computing Machinery, 2023.

[17] (2025) Co-simulating your verilated model. [Online]. Available: https://renode.readthedocs.io/en/latest/tutorials/co-simulating-custom-hdl.html