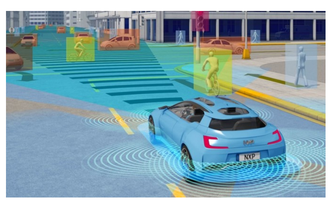
Reliability considerations in the use of high-performance processors in safety-critical systems

PhD Candidate: Sergi Alcaide[1,2]
Thesis Advisors: Leonidas Kosmidis[2], *Carles Hernández*[2,3], *Jaume Abella*[2]

[1]Universitat Politècnica de Catalunya (UPC) Barcelona, Spain
[2]Barcelona Supercomputing Center (BSC) Barcelona, Spain
[3]Universitat Politècnica de València València, Spain

Motivation

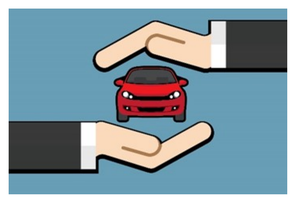
- **Autonomous Driving (AD)** frameworks are too performance demanding to execute on traditional safety-critical platforms.
- Performance levels have been proven to be achieved by many existing Embedded High-Performance Computing (eHPC) platforms.
- Meeting the **safety requirements** of those applications with the highest Automotive Safety Integrity Level (ASIL), as dictated by the automotive functional safety standard, the **ISO26262**, in these more powerful platforms is a **challenge** that must be addressed properly.
- A number of chip vendors already commercialize several processors and platforms for AD systems, **RENESAS R-Car H3** [1] and the **NVIDIA Xavier SoC** [2].
- These platforms did not achieve the highest integrity level certification (**ASIL-D**) which is mandatory in order to be used for AD.



Background

Functional Safety in ISO26262

- Safety critical systems must ensure to be **fault tolerant** since some **faults** cannot be tested and may happen while they are functioning.
- AD must remain **fail operational** in spit of the presence of faults, since ASIL-D functionalities such as braking and steering are managed by it.
- According to ISO26262, this imposes the ASIL-D certification in all the elements included on these functionalities.
- ASIL-D compliance is often achieved by implementing **diverse redundancy** (e.g. Dual-Core Lockstep (DCLS) execution).



ASIL Decomposition

- ASIL decomposition stands for the rules that allow implementing an item with a given ASIL using items with lower ASIL.
- ASIL decomposition is used in the automotive domain to decrease costs by making safety issues become availability issues.
- For AD systems, ASIL decomposition is only applicable by using **diverse redundant** components with some ASIL (e.g. two ASIL-B CPUs), since in case of a failure, system must be **fail-operational** (operation must continue (e.g. decision system).

Contributions

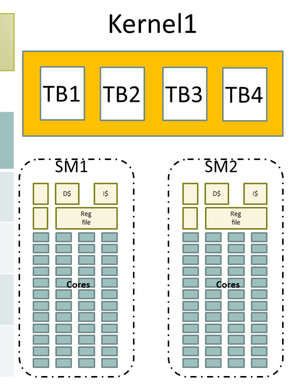
- We focused on **GPUs** because most of the frameworks for AD are defined for GPUs thanks to their efficiency in running parallel algorithms used on image processing and tracking. We divided our GPU contributions in two parts, the ones that required **hardware modification** and the ones that can be implemented by **only-software modifications**, which means that can be implemented on Commercial Off-The-Shelf (COTS) GPUs.
- Both strategies benefit from the GPU **offloading process** which creates an initial staggering for both executions which grants them **time diversity**
- We later focused on **multicores** since they are wide spread in the HPC community and though they do not have the same level of parallelism than GPUs, they have a better single thread performance.

GPU HW contribution [3]:

- Luckily, GPUs have potentially an **internal redundant** scheme thanks to their design. Multiple Streaming Multiprocessors (SMs) are instantiated. Thus, if the scheduler is designed accordingly, redundant work can be split between different SMs.
- Two different schedulers are proposed to better suit different type of kernels, since not all of them can fit in the GPU at the same time:
  1. **Short**: Kernels that are too small to run in parallel
  2. **Friendly**: Kernels that can be executed in parallel
  3. **Heavy**: Kernels that require more than 50% of one resource and cannot be executed in parallel
- **SRRS** is proposed for Short and Heavy kernels, but since friendly can be executed in parallel, **HALF** is also proposed.

SRRS

SM assigned	K1	K'1
TB1	2	3
TB2	3	4
TB3	4	1
TB4	1	2



HALF

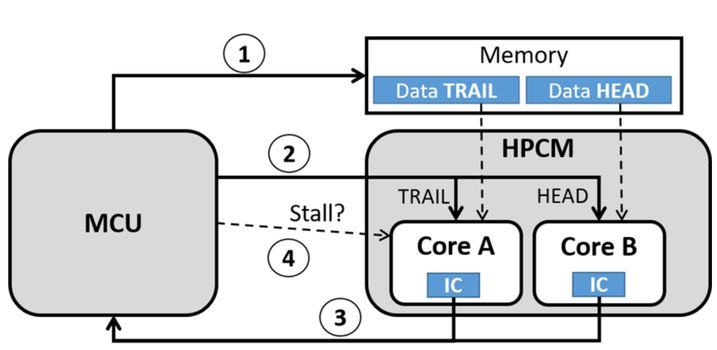
SM assigned	K1	K'1
TB1	1	3
TB2	2	4
TB3	1	3
TB4	2	4

GPU SW contributions [4][5]

- To avoid modification of the HW, we also provided two solutions which implement GPU **dual** and **triple redundancy** by only software means. The evaluation on this part has been done in real **COTS GPU**.

Multicores SW contribution [6]

- By using the **Performance Monitor Counters (PMCs)** which is an already implemented features in most multicores, we set up a dual redundant execution and monitoring the progress of both executions to ensure that a certain distance is maintained during all the execution.
- This is tracked by a Monitor process, which can be implemented in a external ASIL-D microcontroller unit (MCU), that has the power to stall the trail thread in case both executions are too close.



Conclusions & Future Work

In this Thesis we have seen a some of the safety challenges that AD is facing, and we have proposed some solutions based in obtaining a diverse redundant execution in COTS systems. We are still working on the multicores part to allow a parallel execution to take full profit of all the cores available in the system. As a future work, we also expect to have the opportunity to implement our solutions in real hardware in the context of the European Processor Initiative (EPI) project in the automotive part.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under grant TIN2015-65316-P and the HiPEAC Network of Excellence and the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 826647 (EPI).

References

[1] "RENESAS R-Car H3," https://www.renesas.com/enus/solutions/automotive/products/rcar-h3.html.
[2] D. Shapiro, "Introducing Xavier, the NVIDIA AI Supercomputer for the Future of Autonomous Transportation," NVIDIA blog, 2016. [Online]. Available: https://blogs.nvidia.com/blog/2016/09/28/xavier/
[3] S. Alcaide et al. , "High-integrity gpu designs for critical real-time automotive systems," in 2019 Design, Automation Test in Europe Conference Exhibition (DATE), 2019.
[4] S. Alcaide et al. , "Software-only diverse redundancy on gpus for autonomous driving platforms," in 2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS), 2019.
[5] S. Alcaide et al. , "Software-only triple diverse redundancy on gpus for autonomous driving platforms," in 2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), 2020
[6] S. Alcaide et al. , "Software-only based diverse redundancy for asil-d automotive applications on embedded hpc platforms," in [To Appear]The 33rd IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2020.