New Techniques for On-line Testing and Fault Mitigation in GPUs

August 31, 2021

1 Summary

Currently, Graphics Processing Units (GPUs) are crucial devices able to boost the execution of complex algorithms in the scientific and artificial intelligence domains. Moreover, GPU-based platforms are also relevant components now included in several safety-critical applications (e.g., in the automotive and autonomous machines fields), where reliability and functional safety are essential requirements.

The doctoral research activities were focused on identifying new online techniques for testing and mitigation of faults affecting GPUs.

This work also describes a new microarchitectural GPU model (FlexGripPlus), a supporting tool for detailed reliability evaluation. FlexGripPlus can also be used to support the development of functional test approaches and mitigation solutions. This GPU model is compatible with the CUDA programming environment and is a corrected and extended version of a previous GPU model implementing the G80 architecture by NVIDIA. The GPU model's extensions include support for floating-point operations and for the execution of trigonometric and transcendental operations using Special Function Units.

FlexGripPlus was used to develop, evaluate, and validate functional test techniques based on the software-based self-test (SBST) strategy. More in detail, two main strategies are proposed for GPUs: *i)* the multi-kernel test approach, and *ii)* the modular approach of testing.

On the one hand, the multi-kernel approach exploits the GPU's main operative behavior as a special-purpose accelerator. This approach is based on the parallel execution of test programs intended to detect and propagate the fault effects on any GPU core's available outputs. For this purpose, a set of parallel test programs with different GPU configuration parameters allows the generation of test patterns after dividing a target module into fault groups. This division process allows the coverage of sensitive locations that implicitly remain constant by the effect of the configuration of a parallel program. Moreover, this technique also uses a thread-based method to propagate and identify faults into the GPU's available outputs, simplifying the evaluation of specific modules in the GPU architecture, such as the pipeline registers.

On the other hand, the modular testing approach combines the microarchitectural details of a given module, its functional operation, its major constraints, and a target fault model to design a generic description of a feasible test program. More in detail, these GPU features are combined to generate a scalable high-level abstraction test program, which can later be transformed into the equivalent software routines to test a target module. This modular approach exhibited a good effectiveness when testing internal memories and the divergence stack of the GPU core.

It is worth noting that in both strategies (multi-kernel and modular), the combination of high-level programming and low-level assembly language was adopted whenever it was possible. Moreover, several compilation constraints and limitations were listed when implementing the test programs. Both functional test strategies were evaluated targeting the detection of permanent fault models.

Finally, three hybrid and flexible strategies were proposed to harden the GPU architecture. The proposed hardening solution allows the online fault testing and in-field fault mitigation. All three strategies are configurable using custom instructions, so allowing their adoption as part of the code of an application.

The first strategy is based on a flexible approach for in-field fault detection applied to the execution units of the GPU core. This strategy exploits the high regularity of the execution units and provides hardening with reduced overhead costs. The second strategy allows in-field fault mitigation of any functional unit once a fault is detected. In this approach, several in-field configurable spare units are used to provide repair capabilities in the GPU.

The third strategy aims at detecting and mitigating faults at the same time. This flexible approach allows the activation of one or both reliability features. Moreover, the reliability analysis of the proposed strategy showed a considerable increase in the reliability of the targeted units with a minor effect on the running applications code and minimal effect on performance. Furthermore, the implementation of the strategies was evaluated in several configurations, so determining different hardware overhead figures of the proposed hardware mitigation architectures. In the end, these analyzes provide a advisable configuration to obtain maximum reliability benefits with reduced hardware and performance overheads.

Thanks to the availability of the FlexGripPlus model, this work includes for the first time (to the best our knowledge) quantitative results in terms of microarchitectural reliability evaluations aimed at identifying the fault impact of transient faults on several modules of a GPU core, including the scheduler controller, the pipeline registers, the register file, and the branch unit. All these evaluations were performed using several parallel applications. Furthermore, as described above, this model was used as a validation tool in the quantitative evaluation of several online test solutions and hardening strategies for GPUs.

The main results of the research activities are intended to support the development of fault detection and fault-tolerance mechanisms for GPU devices devoted to safety-critical applications, based on the proposed solutions for online

testing and mitigation of faults. Moreover, the experimental results support the evaluation of the different reliability solutions, which are required in the current trends of continuous increment of GPU devices in applications, where the effect of faults is critical.

Finally, as product of the research activities and according to the obtained experimental results, the developed strategies for functional in-field test and in-field mitigation increase the reliability (in up to 50%) and functional-safety (ASIL B) of the GPU, so the combination of the developed strategies can be employed as alternative or complementary fault-tolerance mechanisms for GPUs devoted to applications in the safety-critical domain.