

Software Encoded Processing

Building Safe Systems Despite Unreliable Hardware

Ute Schiffel André Schmitt Martin Süßkraut Christof Fetzer
{ute.schiffel, andre.schmitt, martin.suesskraut, christof.fetzer}@tu-dresden.de

<http://wwwse.inf.tu-dresden.de/>

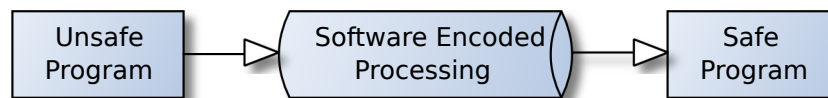
Problem: *Decreasing Reliability of Commodity Hardware*

Future commodity hardware will be much less reliable because of the continuously decreasing feature sizes of integrated circuits¹. However, up-to-date commodity hardware is faster and cheaper than custom reliable hardware. Software Encoded Processing facilitates to build reliable systems using unreliable commodity hardware.

Software Encoded Processing targets mission-critical systems, e.g., safety-critical systems. In these systems hardware errors disturbing the execution of software can have serious consequences. They might result in the loss of human live or in a substantial loss of money.

Solution: *Software Encoded Processing*

In Software Encoded Processing software is automatically transformed from an unsafe original program to a safe version. The functionality of both versions is equivalent. But while the unsafe original version is vulnerable to hardware execution errors, the safe version is with very high probability not. If an execution error disturbs the execution of the safe version, i.e. introduces an erroneous state, this error will be detected. As a consequence, the error can be handled before erroneous output will be produced.



The safe version is obtained by automatically encoding the unsafe version with an Arithmetic Code that facilitates the detection of hardware errors such as disturbances on memory, buses, and in the execution logic. Data is continuously protected from errors during storage, transport, and processing. The usage of Arithmetic Codes allows us to determine the detection probability independent of the characteristics of the hardware used.

Software Encoded Processing provides different safety levels that result in different detection probabilities and different costs in terms of energy, runtime, and memory consumption. These safety levels permit to balance the performance overhead and the error detection coverage.

Gain: *Commodity Hardware for Reliable Systems*

Software Encoded Processing allows to reduce the costs of reliable systems. Cheaper commodity hardware can be used in critical systems.

Since the hardware error detection is implemented in software, the gained flexibility can be used to increase the utilization of systems by executing critical and non-critical applications on the same hardware component.

Software Encoded Processing enables to trade performance and safety. Higher safety levels induce higher costs in terms of energy and runtime. The different levels provided by Software Encoded Processing allow to choose that safety level which delivers the required safety at minimal costs.

¹Shekhar Borkar (Intel Corp.): *Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation* in IEEE Micro, Volume 25, Issue 6 (November 2005), ISSN:0272-1732.