

# Bachelor-/Master Theses

## Approximate Computing in Hardware and Software

Computer architecture is evolving with the workloads. Previous shifts in our workloads have led to introduction of alternative computing platforms such as GPUs and accelerators to address the changing characteristics. Modern applications have different characteristics than before: Error resilience

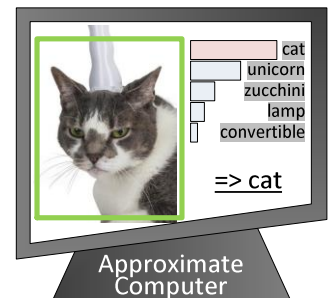
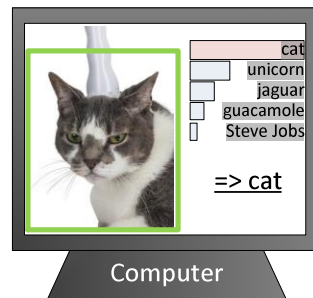
Many application domains including decision making, computer vision, recognition, artificial intelligence (e.g. *deep neural networks*), data mining and synthesis show an intrinsic error tolerance in their computation. This tolerance can be attributed to characteristics such as attenuation of error through statistical methods or iterations. **Approximate computing** leverages application resilience to significantly improve energy and/or performance.

Existing research works have proposed relaxing synchronizations at task level, skipping non-critical computations at software level and reducing circuit precision/complexity or lowering the operation voltage at hardware level. These techniques are becoming an enabling technology for computing heavy workloads with limited resources (e.g. machine learning on smart phones). Commercial examples of reduced precision computing already started to appear (NVIDIA Tegra 1X & Volta, AMD Polaris, Google tensor processing unit, etc.).

Approximate computing has opened many questions to maximize its benefits. We need to answer when, where to apply and also which technique(s) to choose. To answer some of these questions, we offer several theses in different levels and computation layers.

### Potential Thesis Topics:

- Design of hardware accelerators for, e.g., machine learning
- Circuit and  $\mu$ Architecture layer analysis of approximate units.
- Architecture and cross-layer analysis
- Compiler support for approximate computing
- Mixed-signal design & analysis of basic building blocks of processors



### Tasks:

- Tasks will vary according to the thesis topic. In general, they will demand hardware or software development, demonstration with existing benchmarks and analysis

### Skills acquired with the Thesis:

- Hardware / software development and integration skills
- Work in a research environment on current and future topics
- Technical writing
- Prior knowledge on Approximate Computing is **not** required

### Required Knowledge (either-or):

- Verilog/VHDL
- Programming skills
- Compiler background, ideally LLVM
- Cadence Virtuoso

### Helpful skills (not required but helpful):

- Scripting skills (tcl, python)
- Experience with EDA flow
- Analog or digital design experience

### Supervision:

M.Sc. Tanfer Alan

[alan@kit.edu](mailto:alan@kit.edu)

[ces.itec.kit.edu/~alan](http://ces.itec.kit.edu/~alan)

Questions? → Feel free to [contact me](#).