

Design Space Exploration in Chisel

Beim Approximate Computing (AC) wird die Korrektheit von Berechnungen zugunsten geringerer Kosten (z.B. Ausführungszeit, Energieverbrauch) aufgegeben. Um den beste Abwägung zwischen Korrektheit und Dauer/Energieverbrauch zu finden wird eine Entwurfsraumexploration (engl. Design Space Exploration, DSE) durchgeführt.

Mit der modernen Hardware-Beschreibungssprache Chisel bietet einen abstrakteren Zugang zu Hardware, als herkömmliche Sprachen wie VHDL oder Verilog. In ihr lassen sich einfach große Hardware-Projekte (z.B. ein Mehrkern-RiscV-System) umsetzen. Der Chisel-Compiler ist open-source software, die sich leicht erweitern lässt.

Im Rahmen dieser Arbeit soll Chisel/der Chisel-Compiler um ein zu entwickelndes DSE-Framework erweitert werden.

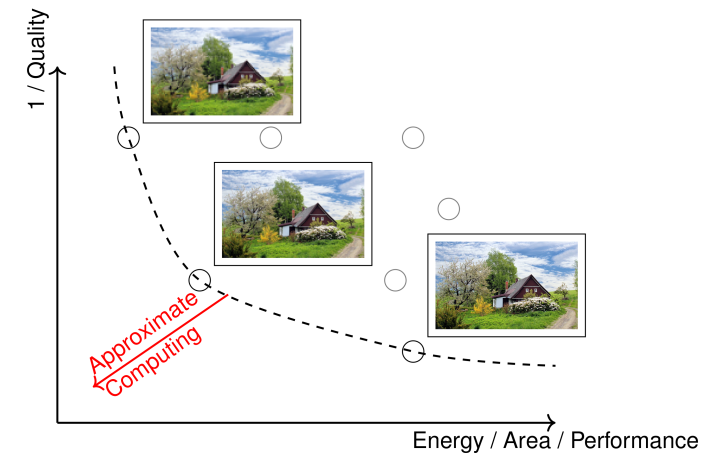
Ziele der Arbeit sind:

- Verständnis von AC + DSE
- Verständnis der Hardware-Beschreibungssprache Chisel
- Implementierung eines DSE-Frameworks in Chisel/Scala

Voraussetzungen: Erfahrung FPGA Programmierung (z.B. Verilog/VHDL) hilfreich, Scala hilfreich

Art der Arbeit: Konzeption (50%), Implementierung (50%)

Betreuer: Oliver Keszöcze (oliver.keszoecze@fau.de)



```
class EdgeDetectionAccelerator(val w: Int, val h: Int)
  extends Module {
    val io = IO(new EdgeAccIO(w,h))
    val image = RegInit(new Image(w,h))

    // detect edges
  }
```