



## Aufgabenstellung Bachelor of Science

### Unterstützung von Fließkommazahlarithmetik in CGRAs in AMIDAR

Institut für Datentechnik  
Fachgebiet Rechnersysteme



#### Voraussetzungen:

1. Praktische Erfahrung im Umgang mit Verilog
2. Grundkenntnisse von Java

#### Einleitung:

Coarse Grained Reconfigurable Arrays (CGRAs) sind Sets aus Rechenzellen, die Instruktionssequenzen massiv parallel verarbeiten können. Die am Fachgebiet Rechnersysteme entworfene CGRA Architektur ist generisch. Es kann mittels eines Generators synthetisierbarer Verilog-Code für jede sinnvolle Zusammenstellung eines CGRAs generiert werden. Zu Beginn dieser Arbeit sind allerdings nur Festkomma-Operationen verfügbar. Um die Einsetzbarkeit des CGRAs zu erweitern, soll dieses auch Fließkomma-Berechnungen durchführen können.

#### Bearbeiter:

Lars Stein  
Matrikel-Nr.: 2010140  
[stein.lars@gmx.de](mailto:stein.lars@gmx.de)

#### Betreuer:

Christian Hochberger  
[hochberger@rs.tu-darmstadt.de](mailto:hochberger@rs.tu-darmstadt.de)

Dennis Wolf  
[wolf@rs.tu-darmstadt.de](mailto:wolf@rs.tu-darmstadt.de)

#### Zeitraumen:

Beginn: 22. Mai 2017  
Ende: 21. Aug. 2017

#### Aufgabe:

Das Ziel der Arbeit ist die Umsetzung der vier Fließkomma-Operationen Addition, Subtraktion, Multiplikation und Division im CGRA. Dabei gilt es den Verilog Generator zu erweitern, dass für jede der vier Operationen synthetisierbarer und getesteter Verilog-Code generiert werden kann. Es sollen beliebige Bitbreiten der Operanden unterstützt werden. Es soll lediglich zwischen Single (32 Bit) und Double (64 Bit) Precision unterschieden werden. Außerdem soll vorgesehen werden, dass die Bitbreite der Operanden größer gleich der Eingangsbreite der ALU sein kann. Dies bedeutet, dass die Übernahme der Operanden mehrere Takte braucht.

Entsprechend der Aufgabenstellung, lassen sich folgende Schritte für den Verlauf der Arbeit ableiten:

- 
1. Recherche zur Fließkommazahlrepräsentation und zu verschiedenen Rechenverfahren für die einzelnen Operationen. Für die Division steht eine Goldschmidt-Implementierung zur Verfügung, die abgewandelt und verwendet werden soll.
  2. Beispielhafte und funktionierende Implementierung einer festen Bitbreite für jede Operation.
  3. Integration in den Generator und Adaption auf verschiedene Bitbreiten. Hierbei muss auch entsprechend der Komposition der CGRAs die Entgegennahme der Operanden bedacht werden.
  4. Für ein sehr gutes Ergebnis sollen belastbare Tests der einzelnen Operationen mit verschiedenen Bitbreiten durchgeführt werden.
  5. Durchführung einer ausführlichen Evaluation der einzelnen Operationen, sowie exemplarischer CGRAs mit Fließkommazahl-Operationen.

Der verfasste Code ist ausführlich zu kommentieren. Modifikationen müssen regelmäßig *committed* und auf dem entsprechenden Branch in das zentrale Git Repository übertragen werden. Die eigene Arbeit bzw. der eigene Branch sollte als eine Bereicherung des UltraSynth Projekts angesehen werden und dementsprechend auch konfliktfrei in den *Master Branch* *merged* werden können.

Außerdem ist eine schriftliche Ausarbeitung im Stil der Rechnersysteme Vorlage [template] anzufertigen. Die Ausarbeitung sollte sowohl die Problemstellung, als auch die geleistete Arbeit, sowie die Resultate reflektieren.

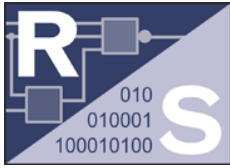
#### **Literatur:**

[template] RS Template: <http://www.rs.tu-darmstadt.de/index.php?id=137>

Prof. Dr.-Ing. Christian Hochberger

Lars Stein

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Institut für Datentechnik Fachgebiet Rechnersysteme

### Erklärung

**Betreff:** Bachelor of Science

**Thema:** Unterstützung von Fließkommazahlarithmetik in CGRAs in AMIDAR

Hiermit erkläre ich, dass ich mit der Nennung meiner Wohn- und E-Mail-Adresse in der PDF-Datei und in der gedruckten Version der Aufgabenstellung des Bachelor of Science einverstanden bin.

Des Weiteren räume ich mit dieser Erklärung der Professur Rechnersysteme ein nicht ausschließliches, übertragbares Nutzungsrecht an der von mir im Rahmen des Bachelor of Science angefertigten Software (Binärprogramme und Quellcode), der entwickelten Hardware und den entworfenen Verfahren ein.

Darmstadt, 22. Mai 2017

Lars Stein

---