

TECHNISCHE UNIVERSITÄT DRESDEN  
FAKULTÄT INFORMATIK  
INSTITUT FÜR TECHNISCHE INFORMATIK  
PROFESSUR FÜR VLSI-ENTWURFSSYSTEME, DIAGNOSTIK UND  
ARCHITEKTUR

## **Bachelorarbeit**

Evaluation einer modernen Zynq-Plattform am Beispiel der  
Implementierung einer Hough Transformation

Dominik Weinrich  
geboren am 11.08.1990 in Kassel  
(Mat.-Nr.: 3914410)

Betreuender Hochschullehrer:  
Prof. Dr.-Ing. habil. Rainer G. Spallek

Betreuer:  
Oliver Knodel

Dresden, Datum



# Aufgabenstellung



# Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Studienarbeit zum Thema

## **Evaluation einer modernen Zynq-Plattform am Beispiel der Implementierung einer Hough Transformation**

selbstständig verfasst und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt, nur die angegebenen Quellen benutzt und die in den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Name, Dresden, Datum

# Wettbewerbsrechtlicher Hinweis

Die bloße Nennung von Namen, Produkten, Herstellern und Firmennamen dient lediglich als Information und stellt keine Verwendung des Warenzeichens sowie keine Empfehlung des Produktes oder der Firma dar.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>Listings</b>	<b>XI</b>
<b>Abkürzungsverzeichnis</b>	<b>XIII</b>
<b>1 Einleitung und Motivation</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>3</b>
2.1 Zielarchitektur - Zynq Systemarchitektur . . . . .	3
2.2 Hardware/Software Codesign . . . . .	3
2.3 High Level Synthese . . . . .	3
2.4 Hough Transformation . . . . .	3
2.4.1 Umwandlung von RGB zu Grayscale . . . . .	3
2.4.2 Gauss Filter . . . . .	3
2.4.3 Canny Edge Detection . . . . .	3
2.4.4 Circle Hough Transformation . . . . .	3
2.4.5 Optimierungen . . . . .	3
<b>3 Hardware/Software Codesign am Beispiel einer Hough Transformation</b>	<b>5</b>
3.1 Softwareimplementierung . . . . .	5
3.2 Iterative Auslagerung einzelner Komponenten auf den FPGA . . . . .	5
<b>4 Evaluation</b>	<b>7</b>
<b>5 Fazit und Ausblick</b>	<b>9</b>
<b>Literaturverzeichnis</b>	<b>i</b>





# Abbildungsverzeichnis



# Tabellenverzeichnis



# Listings



# Abkürzungsverzeichnis

**ABS** Antiblockiersystem

**CPU** Central Processing Unit

**ESP** Elektronisches Stabilitätsprogramm

**FPGA** Field Programmable Gate Array

**GPU** Graphics Processing Unit

**HLS** High Level Synthese

**HW** Hardware

**SW** Software





# 1 Einleitung und Motivation

Vor einigen Jahren hatten die meisten Autos ein ABS<sup>1</sup> und ein ESP<sup>2</sup> zur Unterstützung bei Gefahrenbremsungen und zur Stabilisierung der Fahrt in scharfen Kurven. Mittlerweile ist die Liste der Fahrassistenzsysteme deutlich größer geworden und umfasst unter anderen auch Adaptive Geschwindigkeitsregelanlagen, Spurhalte- und Spurwechselassistenten, Einparkhilfen und Autonome Notbremssysteme. Viele dieser Systeme basieren auf Bildbearbeitungs- und Bildanalysealgorithmen, welche gerade in Gefahrensituationen schnellstmöglich ausgewertet werden müssen. Hierfür ist die Nutzung einer CPU<sup>3</sup> meist keine ausreichende Lösung, da diese zwar eine vergleichsweise hohe Taktrate besitzt, aber aktuell mit bis zu 8 Kernen nicht genug Parallelität bietet, um große Bilder effektiv auswerten zu können. Solche Probleme werden daher vermehrt in Hardware ausgelagert, durch einen FPGA<sup>4</sup> oder eine GPU<sup>5</sup> gelöst.

Diese Bachelorarbeit behandelt eine solche Auslagerung von Software in Hardware. Dazu wird im zweiten Kapitel einleitend die Zielarchitektur vorgestellt und ein Einblick in die Grundlagen des HW<sup>6</sup>/SW<sup>7</sup> Codesigns, so wie der HLS<sup>8</sup> gegeben. Im dritten Kapitel wird am Beispiel einer Hough Transformation eine HLS durchgeführt. Dazu wird zunächst die Softwareimplementierung vorgestellt. Anschließend werden iterativ einzelne Komponenten der Hough Transformation von einer CPU auf einen FPGA ausgelagert.

Das vierte Kapitel behandelt die Auswertung des HW/SW Codesigns hinsichtlich des erbrachten Speedups und des Ressourcenverbrauchs und im fünften Kapitel wird abschließend ein Fazit gezogen und ein Ausblick in das Thema gegeben.

---

<sup>1</sup>Antiblockiersystem

<sup>2</sup>Elektronisches Stabilitätsprogramm

<sup>3</sup>Central Processing Unit

<sup>4</sup>Field Programmable Gate Array

<sup>5</sup>Graphics Processing Unit

<sup>6</sup>Hardware

<sup>7</sup>Software

<sup>8</sup>High Level Synthese



## 2 Grundlagen

### 2.1 Zielarchitektur - Zynq Systemarchitektur

### 2.2 Hardware/Software Codesign

### 2.3 High Level Synthese

### 2.4 Hough Transformation

#### 2.4.1 Umwandlung von RGB zu Grayscale

#### 2.4.2 Gauss Filter

#### 2.4.3 Canny Edge Detection

#### 2.4.4 Circle Hough Transformation

#### 2.4.5 Optimierungen



## 3 Hardware/Software Codesign am Beispiel einer Hough Transformation

### 3.1 Softwareimplementierung

### 3.2 Iterative Auslagerung einzelner Komponenten auf den FPGA



## 4 Evaluation





## 5 Fazit und Ausblick



# Literaturverzeichnis