


Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

Projektdokumentation


„LBP Operator im Vergleich“

Gruppe 5a

Michael Erkel


Simon Friedrich

Sergej Zuyev

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

Inhaltsverzeichnis

1 PROJEKTINITIIERUNG	3
1.1 ARBEITSPAKETE (FÜR DETAILS SIEHE ANHANG ARBEITSPAKETE)	3
1.2 PROJEKTRAHMEN	4
1.2.1 PERSONAL UND RESSOURCEN	4
1.2.2 ERGEBNISSE	5
1.2.3 ZUSTÄNDIGKEITEN	5
1.2.4 ZEITPLÄNE	5
2 PROJEKTPLANUNG	6
2.1 PROJEKTSTRUKTURPLAN	6
2.2 GANTT-DIAGRAMM	7
3 PROJEKTCONTROLLING	8
3.1 MESSVERFAHREN	8
3.1.1 PRÜFPROTOKOLL	9
3.2 PORTIERUNG VHDL-CODE AUF DE1-SOC-BOARD	15
3.2.1 PRÜFPROTOKOLL	16
3.2.2 SIMULATION LBP-KERNEL MIT „SLIDING WINDOW“	17
3.3 OPEN-CL-LÖSUNG	18
3.3.1 PRÜFPROTOKOLL	19
3.4 ÜBERGREIFENDES KOMMUNIKATIONS PROTOKOLL	20
3.5 GANTT-DIAGRAMM MIT VERGLEICH ZUR PROJEKTPLANUNG	22
3 PROJEKTENDE	23
4.1 AUFGETRETENE PROBLEME	23
4.1.1 MATLAB ANWENDUNG	23
4.1.2 VHDL-UMSETZUNG	23
4.1.3 OPEN-CL-UMSETZUNG	24
4.2 RESÜMEE	24

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

1 Projektinitiierung

Nach der gegenseitigen Vorstellung der Teammitglieder studierte jeder für sich die Projektunterlagen, die vom Auftraggeber bereitgestellt wurden (Lasten/Pflichtenheft im Anhang). Gemeinsam im Team diskutierten wir über die Unterlagen und legten die weiteren Arbeitsschritte fest.

Der erste Schritt in der Projektinitiierung ist das Erstellen einer Anforderungsliste. Wir haben hierfür die relevanten Details in den Projektunterlagen auf einem Flipchart zusammengestellt und diese zu Arbeitspaketen zusammengefasst.

1.1 Arbeitspakete (für Details siehe Anhang Arbeitspakete)

AP 1.1: Beschaffung: DE1-SOC-Board

AP 1.2: Einrichtung: Arbeitsumgebung

AP 1.3: Beschaffung: VHDL-Quellcode

AP 1.4: Beschaffung: Masterthesis

AP 1.5: Test: Entwicklungsumgebung

AP 2.1: Analyse: Portierbarkeit des VHDL-Quellcodes auf OpenCL prüfen

AP 2.2: Analyse: Portierbarkeit des VHDL-Quellcodes auf das DE1-SoC-Board prüfen

AP 2.3: Analyse: Ansatz für ein Messverfahren erarbeiten

AP 3.1: Entwicklung: Messverfahren

AP 3.2: Entwicklung: VHDL-Code auf DE1-Soc-Board portieren

AP 3.3: Entwicklung: Umsetzung in OpenCL

AP 4.1: Test: Messverfahren


AP 4.2: Test: VHDL-Umsetzung auf DE1-Soc-Board

AP 4.3: Test: OpenCL-Umsetzung

AP 5.1: Präsentation

AP 5.2: Dokumentation

AP 5.3: Produktvorstellung

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

1.2 Projektrahmen

Nachdem die Arbeitspakete klar definiert waren, wurden nun die groben Rahmengrundlagen für die Realisierung des Projektes festgelegt.

1.2.1 Personal und Ressourcen

Personal:

- Michael Erkel
- Simon Friedrich
- Sergej Zuyev

Ressourcen:

Software:


- Altera Softwareumgebung
- Linux-Distribution für OpenCL
- Mathworks Matlab R2012a

Hardware:

- DE1-SOC-Board
- Entwicklungs/Präsentationrechner mit Peripherie
- Micro-SD-Speicherkarte mit mindestens 4 GB

Unterlagen:

- Vollständige Dokumentation von DE1-SOC-Board
- Masterarbeit Herr Kumar
- Lastenheft
- Pflichtenheft

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

1.2.2 Ergebnisse

- Messverfahren ist entwickelt
- VHDL-Code wurde auf DE1-SOC-Board portiert
- Open-CL-Lösung ist entwickelt
- Dokumentation ist erstellt
- Präsentation ist erstellt

1.2.3 Zuständigkeiten

Michael Erkel:

- VHDL-Code auf DE1-SOC-Board portieren
- Dokumentation erstellen

Simon Friedrich:


- Messverfahren entwickeln
- Präsentation erstellen

Sergej Zuyev:

- Open-CL-Lösung entwickeln

1.2.4 Zeitpläne

1. Einrichten der Entwicklungsumgebung	01.08.2017 – 09.08.2017
2. Analyse	10.08.2017 – 18.08.2017
3. Entwicklung	19.08.2017 – 09.09.2017
4. Testen	10.09.2017 – 11.09.2017
5. Projektabschluss	12.09.2017 – 21.09.2017

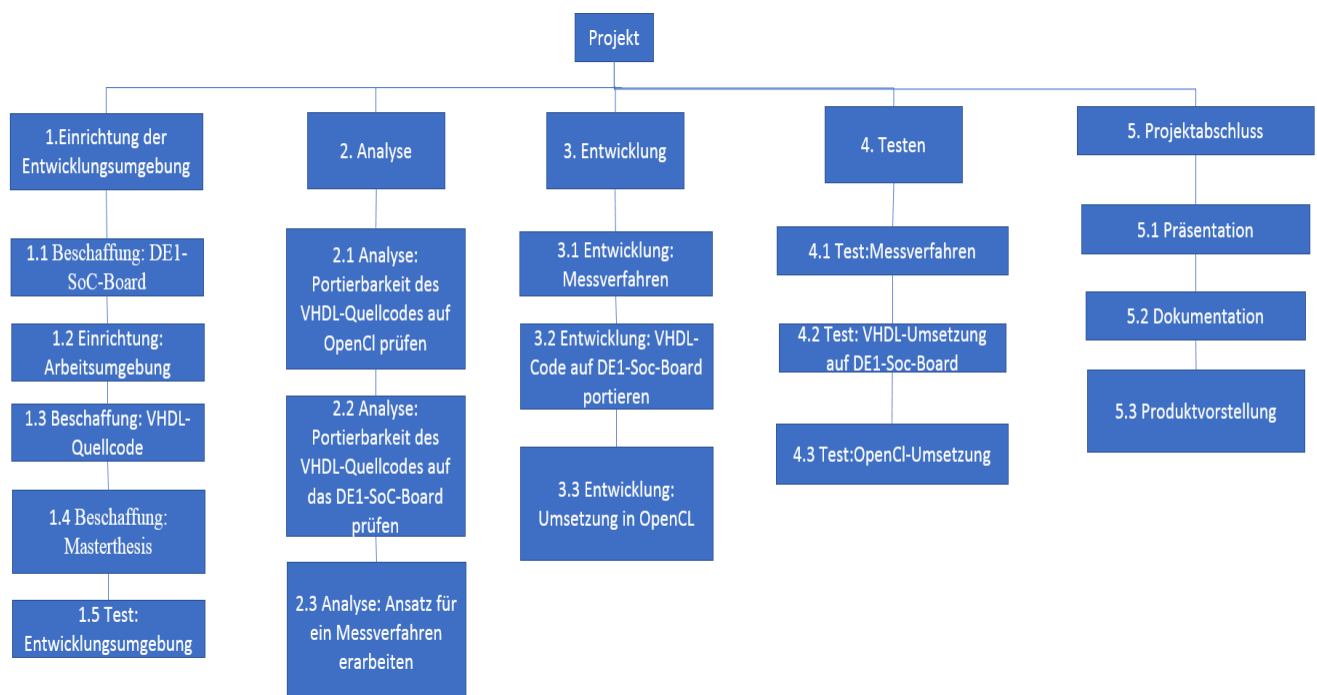
Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	


2 Projektplanung

Im nächsten Schritt ging es weiter mit der Projektplanung. Die Vorgehensweise wurde detaillierter geplant und die Zeiten und Ressourcen in den Arbeitspaketen nachgepflegt.

2.1 Projektstrukturplan

Abgeleitet aus den Arbeitspaketen wurde folgendes PSP erstellt.

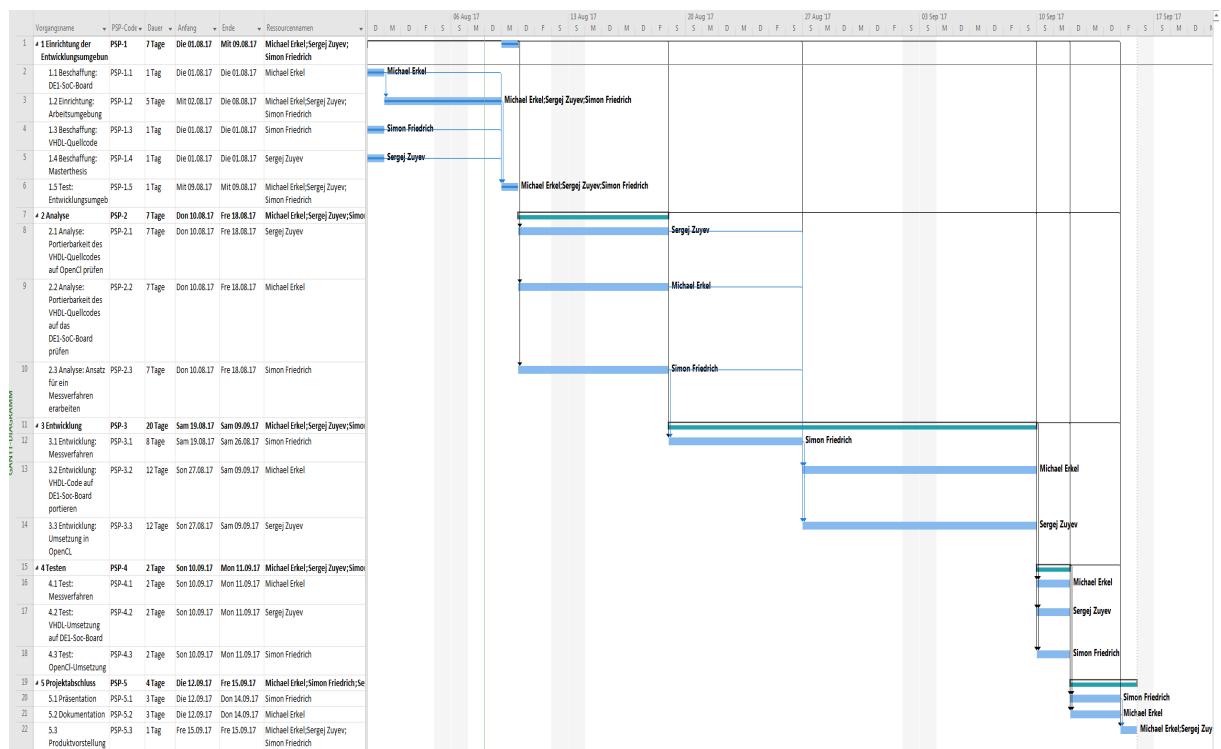



Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

2.2 Gantt-Diagramm

Danach haben wir uns auf den zeitlichen Ablauf des Projektes geeinigt, welchen wir mit Hilfe des Gantt- Diagramm dargestellt haben. Im Diagramm sind die Sammel- und Einzelvorgänge zu sehen und deren zeitliche Einteilung.

Des weiteren ist im Diagramm die Ressourcenplanung zu erkennen.



Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3 Projektcontrolling

Nun folgte das Projektcontrolling.

Nach jedem Softwareabschnitt haben wir anhand unserer Prüfprotokolle den Projektstatus überprüft.


3.1 Messverfahren

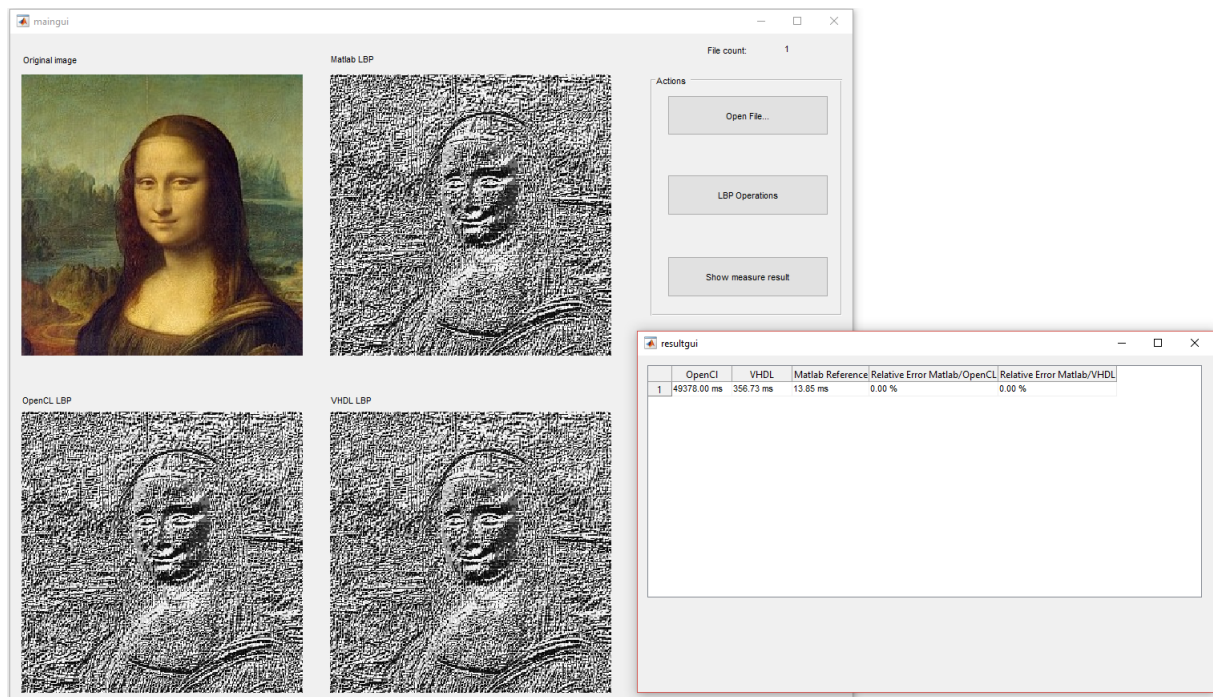
Grundüberlegung:

Zur Umsetzung des Benchmark-Tests ist ein Messverfahren der Verarbeitung durch den LBP-Operator von Nöten. Weiterhin wird eine Möglichkeit der Darstellung benötigt.

Umsetzung:

Es wird eine MATLAB GUIDE Anwendung entwickelt. Diese bietet diverse Möglichkeiten der Darstellung (bspw. Tabellarisch, als Diagramm, etc.), sowie einfache Verfahren zur Messung der unterschiedlichen Lösungen. Als Zusatz kann ein Vergleich der Ergebnisse mit der MATLAB-Lösung des LBP-Operators von Herrn Kumar angestellt werden. Zur konkreten Messung der Verfahren wird eine Zeitmessung inklusive der Datenübertragung an das Board durchgeführt. In Abhängigkeit des Aufwandes wird optional eine Zeitmessung ohne Datenübertragung umgesetzt. Die Anwendung soll ebenfalls Dateien (Bilder) öffnen können und diese zur Vorbereitung in Graustufen umwandeln. Zudem werden die umgewandelten Daten von der Anwendung an das DE1-SoC Board gesendet.

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	




3.1.1 Prüfprotokoll

1. Testfall

Dateiname1: IMG0.jpg

Dateiname2: IMG0.jpg

Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

2. Testfall

Dateiname1: IMG1.jpg

Dateiname2: IMG1.jpg


Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

3. Testfall

Dateiname1: IMG2.jpg

Dateiname2: IMG2.jpg

Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

4. Testfall

Dateiname1: IMG3.jpg

Dateiname2: IMG3.jpg


Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

5. Testfall

Dateiname1: IMG4.jpg

Dateiname2: IMG4.jpg

Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

6. Testfall

Dateiname1: IMG5.jpg

Dateiname2: IMG5.jpg


Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

7. Testfall

Dateiname1: IMG6.jpg

Dateiname2: IMG6.jpg

Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	


Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

8. Testfall

Dateiname1: IMG7.jpg

Dateiname2: IMG7.jpg

Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

9. Testfall

Dateiname1: IMG8.jpg

Dateiname2: IMG8.jpg


Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

10. Testfall

Dateiname1: IMG9.jpg

Dateiname2: IMG9.jpg


Plattform	Dateien konnten Fehlerfrei eingelesen werden		Dateien werden dargestellt		Übereinstimmung in %	Kommentar
	ja	nein	ja	nein		
VHDL		x		x	0	Nicht lauffähig
OpenCL	x		x		N/A	FPGA: falsch, GPU: visuell korrekt
Matlab (Referenz)	x		x		100	

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3.2 Portierung VHDL-Code auf DE1-Soc-Board

Grundüberlegungen:

- LBP Operator – Verechnet ein 3x3 Datenfenster
- LBP-Kernel - Verarbeitet Daten in einem Speicherbereich
- LBP Kernel Array - Mehrere Kernel
- Processing unit
 - Hauptrecheneinheit, beinhaltet:
 - Eingabepuffer
 - Ausgabepuffer
 - Kernel array
- Processing unit input memory
 - Eingabepuffer, Speicherbereich für das komplette Eingangsbild
 - Block-RAM basiert
 - Multi-port fähig
- Processing unit output memory
 - Ausgabepuffer, Speicherbereich für das verarbeitete Ausgangsbild
 - Block-RAM basiert
 - Bedingt Multi-port fähig
- Control unit
 - Hauptsteuereinheit
 - Zuständig für Datentransfers und Steuerung vom Host
 - Implementiert das oben beschriebene (Matlab -> ZedBoard) Protokoll
- Host interface UART (PHY)
 - UART Schnittstelle
- Avalon Schnittstelle
 - Kommunikation mit dem ARM Prozessor über die Avalon / AXI Schnittstelle

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

Umsetzung:


- Für das DE1-SoC ist der Aufwand enorm:
 - Avalon MM Schnittstelle, damit eine Anbindung an den ARM Hauptprozessor stattfinden kann
 - Linux-Kernelmodul als Treiber
 - Userspace-Programm für den eigentlichen Datentransfer (Ansteuerung vom Host via SSH, vgl. dazu DatenTransfer für OpenCL)
- Es wurde ein Qsys-Projekt erstellt
 - LBP-modul mit Avalon Schnittstelle als IP-Core
 - Hard Processor System IP-Core
- Umsetzung konnte nicht komplett abgeschlossen werden.

3.2.1 Prüfprotokoll

2. Vergleichbarkeit der einzelnen Messungen


Plattform:	Messergebnis in ms (s für OpenCL)										Mittelwert In ms
	1	2	3	4	5	6	7	8	9	10	
Matlab	5	5	4	6	4	5	4	4	4	4	4.5
VHDL	0	0	0	0	0	0	0	0	0	0	0
OpenCL	46	46	46	46	46	46	46	46	46	46	46000

	OpenCL	VHDL	Matlab Reference	Relative Error Matlab/OpenCL	Relative Error Matlab/VHDL
1	4.6325e+07	0	4.4812	85.6408	0
2	4.6325e+07	0	5.1301	119.9578	0
3	4.6325e+07	0	4.1136	117.4096	0
4	4.6325e+07	0	5.5395	114.3964	0
5	4.6325e+07	0	4.3922	83.9575	0
6	4.6325e+07	0	5.0703	132.9779	0
7	4.6325e+07	0	4.1583	117.3042	0
8	4.6325e+07	0	4.2773	142.0392	0
9	4.6325e+07	0	3.6675	153.1062	0
10	4.6325e+07	0	3.8917	140.9976	0

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3.2.2 Simulation LBP-Kernel mit „Sliding Window“



Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3.3 Open-CL-Lösung

Umsetzung:


2 Varianten wurden entworfen. Beide wurden zuvor auf dem Rechner getestet (Intel Core i7 7700K, 32GB DDR4 RAM, NVidia GeForce GTX1080Ti)

- Die erste Variante war nicht lauffähig aufgrund von Überschreitung der FPGA-Ressourcen. Wird im Kapitel 4.1.3 genauer beschrieben.
- Die zweite Variante wurde stark vereinfacht, dadurch konnte das Ressourcenproblem umgangen werden.
 - Anzahl Samples wurde auf 8 fixiert
 - Radius diskret
 - Keine Gleitkommaoperationen
 - Keine trigonometrischen Operationen

Zur Beschleunigung des Entwicklungs/Testworkflows wurde ein Programm in Python entwickelt und der jeweilige Kernel auf der Grafikkarte des Entwicklungsrechners ausgeführt. Dies ist der Tatsache geschuldet, dass ein kompletter Syntheselauf mit dem Altera OpenCL Compiler ca. 20-30 Minuten in Anspruch nimmt. Mit dem entwickelten Python-Programm kann die Korrektheit des Algorithmus in Sekundenschnelle nachgewiesen werden.

Ergebnis:

Die 2. Variante kann verwendet werden und belegt 19-20% der Ressourcen. Es können mehrere Compute Units synthetisiert werden.


Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3.3.1 Prüfprotokoll

2. Vergleichbarkeit der einzelnen Messungen

Plattform:	Messergebnis in ms (s für OpenCL)										Mittelwert In ms
	1	2	3	4	5	6	7	8	9	10	
Matlab	5	5	4	6	4	5	4	4	4	4	4.5
VHDL	0	0	0	0	0	0	0	0	0	0	0
OpenCL	46	46	46	46	46	46	46	46	46	46	46000

	OpenCL	VHDL	Matlab Reference	Relative Error Matlab/OpenCL	Relative Error Matlab/VHDL
1	4.6325e+07	0	4.4812	85.6408	0
2	4.6325e+07	0	5.1301	119.9578	0
3	4.6325e+07	0	4.1136	117.4096	0
4	4.6325e+07	0	5.5395	114.3964	0
5	4.6325e+07	0	4.3922	83.9575	0
6	4.6325e+07	0	5.0703	132.9779	0
7	4.6325e+07	0	4.1583	117.3042	0
8	4.6325e+07	0	4.2773	142.0392	0
9	4.6325e+07	0	3.6675	153.1062	0
10	4.6325e+07	0	3.8917	140.9976	0

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3.4 Übergreifendes Kommunikationsprotokoll

Zusätzlich zu den beschriebenen Arbeitspaketen musste ein Kommunikationsprotokoll zwischen Host-Rechner und dem DE1-SoC Board entworfen werden. Da das DE1-SoC Board mit einem Linux-Betriebssystem präpariert wurde, lag es nahe ein Protokoll auf hoher Ebene zu verwenden. Hierzu fiel die Wahl auf SSH (Secure Shell) zur Befehlsausführung und SCP (Secure CoPy) für bidirektionale Datentransfers. Ein UNIX-basierendes Betriebssystem gilt als Voraussetzung.

1. Datentransfer zum Board: **ttd** (Transfer To Device)

scp input.dat root@192.168.0.123:/ocl_data

Über SCP wird eine Datei mit Luminanzdaten auf das Board transferiert, danach kann die Berechnung erfolgen.

2. Ausführung auf dem Board **cmd** (Command), bestehend aus 3 Unterbefehlen:

- Initialisierung der OpenCL Umgebung: *source ./init_opencl.sh*
- Programmierung des FPGA: *aocl program /dev/acl0 lbp_ocl.aocx*
- Ausführung des Hostprogramms mit Datei input.dat, Breite = 256, Höhe = 256, Radius = 1 : *./lbp_ocl_host input.dat 256 256 1*
- Gesamtbefehl: *ssh root@192.168.0.123 "source ./init_opencl.sh;aocl program /dev/acl0 lbp_ocl.aocx;./lbp_ocl_host input.dat 256 256 1"*
- Ergebnis wird unter input.dat.res gespeichert


3. Datentransfer zum PC: **tth** (Transfer To Host)

scp root@192.168.0.123:/ocl_data/input.dat.res input.dat.res

Über SCP (Secure Copy) wird eine Datei mit Ergebnissen zurück transferiert, diese beinhaltet:

- Kernelzeit (Reine Ausführungszeit)
- Systemzeit (Kernelzeit + Datentransfer vom Hostprogramm zum Kernel)
- LBP-Verarbeitete Bilddatei (Luminanzdaten)

Außerdem musste ein Low-Level-Protokoll entwickelt werden, welches eine Kommunikation mit dem in VHDL implementierten LBP Operator erlaubt. Dasselbe Protokoll sollte jeweils für die Avalon MM (DE1-SoC) und die UART (ZedBoard) Schnittstellen verwendet werden. Dieses Protokoll soll programmseitig im Kernelmodul (Treiber) für das DE1-SoC bzw. in der Matlab-Anwendung für das ZedBoard umgesetzt werden.


Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

Es werden 4 Befehle und eine Empfangsbestätigung **ack** interpretiert.

S ist hierbei der Slave (LBP-Modul).

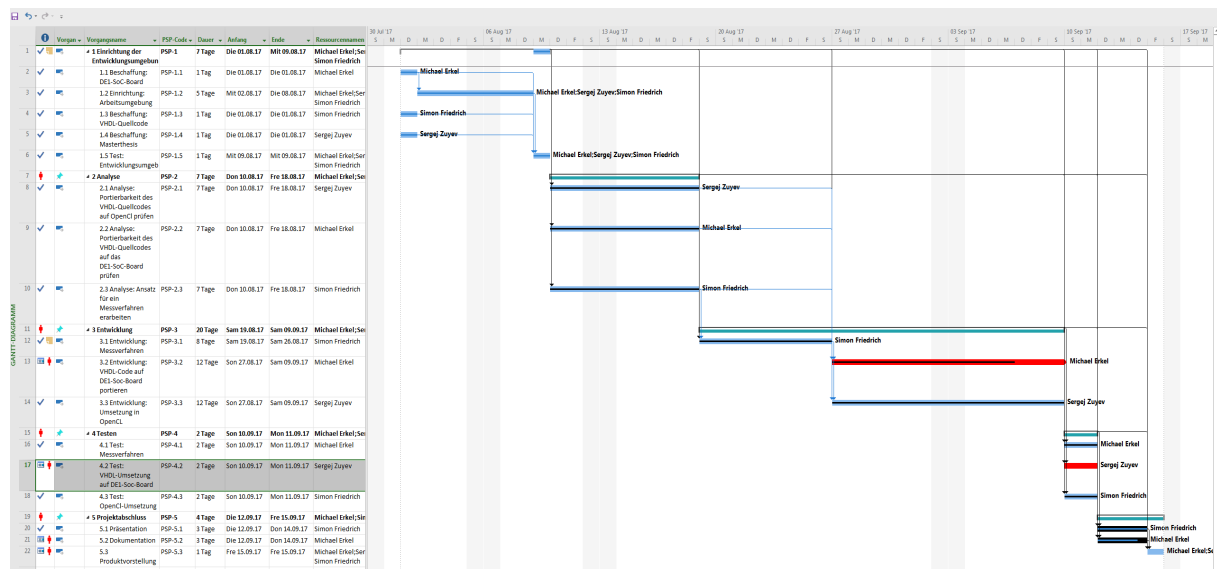
M ist hierbei der Master (Kernelmodul / Matlab Anwendung)


1. Initialisierung **init**
 - a. $M \rightarrow \text{init} \rightarrow S$ (Device wird initialisiert)
 - b. $M \leftarrow \text{ack} \leftarrow S$ (Host darf Daten übertragen)
2. Datentransfer zum Board **ttd** (Transfer To Device), für jedes zu übertragende Byte:
 - a. $M \rightarrow \text{ttd} \rightarrow S$
 - b. $M \leftarrow \text{ack} \leftarrow S$
 - c. $M \rightarrow xhi \rightarrow S$ (Obere 8 bit der Spaltenadresse)
 - d. $M \leftarrow \text{ack} \leftarrow S$
 - e. $M \rightarrow xlo \rightarrow S$ (Untere 8 bit der Spaltenadresse)
 - f. $M \leftarrow \text{ack} \leftarrow S$
 - g. $M \rightarrow yhi \rightarrow S$ (Obere 8 bit der Zeilenadresse)
 - h. $M \leftarrow \text{ack} \leftarrow S$
 - i. $M \rightarrow ylo \rightarrow S$ (Untere 8 bit der Zeilenadresse)
 - j. $M \leftarrow \text{ack} \leftarrow S$
 - k. $M \rightarrow \text{dat} \rightarrow S$ (ein Datenbyte)
 - l. $M \leftarrow \text{ack} \leftarrow S$ (Host darf erneut einen neuen Befehl ausführen)
3. Ausführung auf dem Board **proc** (Process)
 - a. $M \rightarrow \text{proc} \rightarrow S$
 - b. $M \leftarrow \text{ack} \leftarrow S$ (Verarbeitung begonnen)
 - c. $M \leftarrow \text{ack} \leftarrow S$ (Verarbeitung beendet)
4. Datentransfer zum PC: **tth** (Transfer To Host), für jedes zu übertragende Byte:
 - a. $M \rightarrow \text{tth} \rightarrow S$
 - b. $M \leftarrow \text{ack} \leftarrow S$
 - c. $M \rightarrow xhi \rightarrow S$ (Obere 8 bit der Spaltenadresse)
 - d. $M \leftarrow \text{ack} \leftarrow S$
 - e. $M \rightarrow xlo \rightarrow S$ (Untere 8 bit der Spaltenadresse)
 - f. $M \leftarrow \text{ack} \leftarrow S$
 - g. $M \rightarrow yhi \rightarrow S$ (Obere 8 bit der Zeilenadresse)
 - h. $M \leftarrow \text{ack} \leftarrow S$
 - i. $M \rightarrow ylo \rightarrow S$ (Untere 8 bit der Zeilenadresse)
 - j. $M \leftarrow \text{dat} \leftarrow S$ (ein Datenbyte, Host darf erneut einen neuen Befehl ausführen)

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3.5 GANTT-Diagramm mit Vergleich zur Projektplanung

Die Themen von Herrn Erkel konnten nicht beendet werden.
Eine Übergabe an Herrn Zuyev ist erfolgt.



Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

3 Projektende

Zum Abschluss des Projekts wurde im Schritt Projektende eine digitale Präsentation erstellt. Mit dieser stellen wir unsere Ergebnisse des Projektes den anderen Teams vor.


4.1 Aufgetretene Probleme

4.1.1 MATLAB Anwendung

Während der Entwicklung der Anwendung gab es nur wenige kleine Probleme. Diese begrenzten sich meist auf Schwierigkeiten in der Umsetzung, wie beispielsweise die Formatierung der Ergebnisse in Tabellenform. Eine weitere Schwierigkeit stellte die Einbindung des LBP Operators von Herrn Kumar dar, da der gegebene Quellcode nur wenig kommentiert und auch sonst nicht viele Informationen zur Programmlogik vorhanden waren.

4.1.2 VHDL-Umsetzung

- Aufgrund von Fehlern bei der Umsetzung des LBP-Moduls nicht synthetisierbar
- Potentielle Probleme mit dem Betriebssystem / Treiber falls zur Laufzeit das FPGA umprogrammiert wird (Wechsel zwischen VHDL / OpenCL Lösung)
- Auf eine weitere Entwicklung wurde verzichtet und ein reiner Hardware-basierter Ansatz auf Basis des ZedBoards (um Rekonfiguration bei der Präsentation zu vermeiden) mit UART-Schnittstelle weiterverfolgt


Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

4.1.3 Open-CL-Umsetzung

- viele Gleitkomma Operationen
- Ein Aufruf wie z.B round(3.45) ergab statt 3.0 einen nicht nachvollziehbaren Wert, was innerhalb der Indexberechnung zu Speicherbereichsüberschreitungen führt. Dieses Problem trat bei der Ausführung auf der GTX1080Ti GPU auf und wurde durch Ersetzung mit einer eigenen Lösung umgangen.
- nicht synthetisierbar
 - Gleitkommaarithmetik und trigonometrische Funktionen
 - Sehr große generierte "Schaltung"
 - Passt nicht auf das Cyclone V des DE1-SoC: 97% aller Logikzellen belegt
- Neuentwicklung des Kernels, Reduktion auf das Wesentliche

4.2 Resümee

- Mit OpenCL ist ein Ergebnis deutlich schneller zu erzielen
- OpenCL heißt nicht gleich OpenCL. Für die Hardwaresynthese müssen viele Details beachtet werden wie z.B Vermeidung von Gleitkommaoperationen.
- Die Ergebnisse der OpenCL-Lösung, ausgeführt auf der Grafikkarte und dem FPGA, unterscheiden sich enorm trotz gleichem Quellcode.
- Die OpenCL-Lösung ist auf dem DE1-SoC sehr langsam
- Die VHDL-Lösung ist nicht lauffähig
- Die VHDL-Lösung bereitet einen sehr viel höheren Entwicklungsaufwand

Projektname:	LBP Benchmark zwischen Software- und Hardwarelösung	
Dokument:	Dokumentation (Version 0)	
Bearbeiter:	Michael Erkel Simon Friedrich Sergej Zuyev	
Team-Nr.:	5a	
Datum:	20.09.2017	

Änderungshistorie		
Version	Art der Änderung	Datum
0	Erstausgabe	20.09.2017

Anhänge:
Lastenheft
Pflichtenheft
Arbeitspakete