

ERA Praktikum SS 2017
Gleitkomma-Arithmetik
Realisierung von $\sin x$

ERA Praktikum SS 2017	1
1. Einleitung	3
2. Realisierung der Sinus-Funktion in Assembler	3
2.1 Aufgabenkurzbeschreibung	3
2.2 Ist-Zustand	3
2.3 Soll-Zustand	3
2.4 Werkzeug	4
2.5 Aufgabenverteilung	4
3. Organisation	4
3.1 Zeitplanung	4
3.2 Abgabefristen	5
4. Vertragsverpflichtung	5

1. Einleitung

Im Rahmen des ERA-Praktikums an der TU München sollen zwei Projekte in einem Team von drei Studierenden bearbeitet werden.

Das Team besteht aus Thea Kramer, Florian Müller und Berzan Yildiz.

Die zwei zu bearbeitenden Projekte sind:

- I. Realisierung der Sinus-Funktion in Assembler
- II. Entwicklung eines Seriell/Parallel-Konverters in VHDL

Dies ist das Pflichtenheft zum ersten Projekt.

2. Realisierung der Sinus-Funktion in Assembler

2.1 Aufgabenkurzbeschreibung

Ziel ist es, unter Berücksichtigung verschiedener Implementationsweisen die Sinusfunktion *float sin(float x)* in Assembler umzusetzen.

Zur Überprüfung und Durchführung einer Leistungsmessung soll ein C-Rahmenprogramm zur Verfügung gestellt werden.

2.2 Ist-Zustand

Zur Verfügung stehen die vier Grundrechenarten als FPU-Befehle sowie Informationen zur Annäherung der Sinusfunktion mithilfe der Taylor-Mac Laurin'schen Formel.

2.3 Soll-Zustand

Es sollen ausschließlich die vorhandenen FPU-Befehle für die vier Grundrechenarten sowie die notwendigen Speicherverwaltungs- und Kontrollbefehle genutzt werden.

Zur Umsetzung werden zwei Ansätze in Betracht gezogen, einerseits eine Lookup-Table und andererseits eine Annäherung durch eine Reihenentwicklung (z.B mit der Taylor-Mac Laurin'schen Formel).

Für letzteres müssen zunächst Unterprogramme zur Berechnung von Fakultät und Potenz implementiert werden.

Zum Validieren der Funktion durch das C-Rahmenprogramm soll ein Vergleich mit einer geeigneten Referenz (z.B FPU-Befehl oder eine entsprechende C-Routine) durchgeführt werden. Außerdem soll ein Leistungsvergleich in Form einer Zeitmessung stattfinden.

Im C-Code soll die Assemblerfunktion ausschließlich aufgerufen werden (kein Inline-Assembler).

Abschließend soll zur Kompilierung ein Makefile erstellt werden.

2.4 Werkzeug

Zur Implementierung in Assembler wird nasm verwendet. Der Test wird mithilfe eines C-Rahmenprogramms realisiert, in welchem die Assembler-Routine aufgerufen wird. Anschließend wird eine Makefile erstellt, welche dieses mithilfe des GNU C-Compilers kompiliert.

2.5 Aufgabenverteilung

- Projektleitung: Berzan Yildiz
- Dokumentation: Florian Müller
- Vortrag: Thea Kramer

3. Organisation

Der Projektleiter Berzan Yildiz übernimmt in diesem Projekt die Verantwortung für die Hauptorganisation. Er fertigt die Protokolle für jedes Treffen an und lädt sie anschließend in das Repository hoch.

Die Dokumentation wird von Florian Müller geschrieben und die abschließende Präsentation von Thea Kramer gehalten.

3.1 Zeitplanung

Aufgabe	Berzan Yildiz (Projektleiter)	Florian Müller (Dokumentation)	Thea Kramer (Vortrag)	Gesamt
Organisation	1h	0h	0h	1h
Besprechungen	12h			12h
Aufgabenanalyse	4h			4h
Pflichtenheft	4h			4h
Lösungsansätze	6h			6h
Spezifikation	7h			7h
Implementierung	11h			11h
Dokumentation	0h	5h	0h	5h
Vortrag	1h	0h	5h	6h
Protokollierung	4h	1h	1h	6h
Gesamt	50h	50h	50h	62h

3.2 Abgabefristen

Pflichtenheft	28.05.2017
Spezifikation	18.06.2017
Implementierung	09.07.2017
Ausarbeitung	23.07.2017
Vortrag	07.08 - 18.08.2017

4. Vertragsverpflichtung

Im Rahmen dieses Projektes werden von den Teammitgliedern folgende Dokumente angefertigt und über das SVN-Repository abgegeben:

- Pflichtenheft
- Spezifikation
- Implementierung in nasm und Makefile-Erstellung gcc
- Ausarbeitung (Benutzer- und Anwenderdokumentation)

Zudem wird am Ende ein Abschlussvortrag zur Vorstellung des Projekts gehalten.