

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

ERA Praktikum SS 2017	1
Ablauf des Projekts, Probleme und Lösungen	3
Rückblickende Analyse für Umsetzung der Zeitplanung	4
Allgemeine Bewertung der Teamarbeit	4

Ablauf des Projekts, Probleme und Lösungen

Wir haben in unserem ersten Treffen uns einen groben Überblick über die Aufgabenstellung gemacht und unsere Rollen verteilt, und zwar mich (Berzan Yildiz) als Projektleiter, Florian Müller für die Dokumentation und Thea Kramer für den Vortrag.

Im nächsten Treffen haben wir das Pflichtenheft fertiggestellt und uns im Rahmen dessen mit den zu erwägenden Lösungsmethoden (Reihenentwicklung und Lookup-Table) und Werkzeugen zur Implementierung (nasm, C-Rahmenprogramm und zu der Zeit noch voraussichtlich GNU) auseinandergesetzt.

Anschließend haben wir die Spezifikation fertiggestellt, wobei wir uns gegen die Reihenentwicklung und für die effiziente Lookup-Table entschieden haben, und auch direkt in Pseudocode festgehalten haben, wie wir diese verwirklichen würden. Unser Schema, zunächst den Eingabewert x auf $\frac{\pi}{2}$ abzubilden, dann Binär mithilfe dieser Abbildung in der Lookup-Table gesucht wird und zuletzt zu interpolieren, hat sich tatsächlich bis zur endgültigen Version gehalten.

Wir haben dann direkt angefangen, den Pseudocode in Assemblercode umzuschreiben, wobei vor allem die Binäre Suche am kompliziertesten war. Wir sind an einem Tag bis zu der Interpolation gekommen.

Zum nächsten Treffen ist uns jedoch aufgefallen, dass wir alles mit den regulären Registern implementiert hatten, und eigentlich die FPU nutzen mussten. Somit haben wir einen ganzen Tag an Zeit verloren, letztendlich jedoch alles fertigstellen können. Die Implementation mithilfe der FPU war dadurch schwierig für uns, dass wir im ersten Semester nicht in dieser Weise gearbeitet hatten und uns somit komplett in die Herangehensweise von FPU-Assembler einarbeiten mussten. Außerdem haben wir kleinere Denkfehler wie überschätzte Speicherplatzgrößen, oder falsche Bedingungen für Sprünge behoben.

Im nächsten Treffen haben wir uns an die Kompilierung gesetzt. Die Einarbeitung in die Tools und Arbeitsweise mit Assembler und C hat uns sehr viel Zeit und Aufwand gekostet. Wir haben alles auf MacOS gemacht, da uns schnell aufgefallen ist, dass die Arbeit mit einem Unix-System erheblich einfacher ist und bei den Windows-Systemen oft Fehler auftraten.

Um Fehler im Code zu debuggen, haben wir immer einzelne Abschnitte kompiliert, auf entstehende Fehler geachtet und uns somit Abschnittsweise voran getastet, bis keine Fehler mehr vorlagen. Bei dem Schreiben des C-Rahmenprogramms mussten wir dann auch einige MacOS spezifische Dinge wie die Nutzung von Clang statt GNU oder verschiedenen Funktionennamen in C (z.B. `__floatsin` statt `floatsin`) beachten. Die Übergabe eines Wertes von C nach Assembler war definitiv der schwierigste Teil bei dem C-Rahmenprogramm, dies war jedoch mit etwas Recherche lösbar.

Im Großen und Ganzen war die größte Schwierigkeit hierbei definitiv, dass wir nicht normal Debugger konnten, und dass man nicht so leicht Antworten auf jegliche Fragen und Fehler im Internet findet, da es sehr wenig C und Assembler bezogene Beiträge und Referenzen gibt, zumindest im Vergleich zu Java.

Das Schreiben des Testprogramms war nicht weiter schwierig, da C sich relativ ähnlich zu Java verhält (außer den Eigenarten von "scanf" und "printf", welche an manchen Stellen gestört haben) und auch die Einbindung der Standardbibliothek für die Sinusfunktion unkompliziert war. Auch das Erstellen der Makefile war nicht weiter schwierig.

Rückblickende Analyse für Umsetzung der Zeitplanung

Rückblickend war die insgesamt eingeplante Zeit nicht schlecht eingeschätzt, denn bei 8 Treffen mit im Schnitt 7 Stunden Arbeit kamen wir auf ca. 56 Stunden pro Person, Vortrag und Dokumentation nicht mit einberechnet.

Die Organisation hat nicht mehr als 1 Stunde in Anspruch genommen. Auch die Besprechungen und die Aufgabenanalyse waren gut vorgeplant, hier sind wir nicht stark von abgekommen.

Das reine Schreiben des Pflichtenheft hat auch nicht wirklich mehr als 4 Stunden gedauert. Insgesamt war die theoretische Arbeit und Zeit zum Nachdenken gut geplant, hier kamen 6 Stunden insgesamt gut hin.

Für die Spezifikation haben wir etwas mehr als für das Pflichtenheft gebraucht, auch hier kommt die Planung hin. Da wir immer Assembler vor VHDL bearbeitet haben, ist hier auch ein Großteil der Vorarbeit für die Strukturierung der Dokumente geschehen.

Die Implementierung insgesamt hat uns definitiv mehr als 11 Stunden gekostet, eher näher an 14. Sie war definitiv der aufwendigste und auch anspruchsvollste Teil der ganzen Arbeit.

4 Stunden für die Protokollierung war auch stark überzogen, denn ein Protokoll zu schreiben hat maximal 20 Minuten gedauert.

Im Großen und Ganzen also haben wir unsere Zeitplanung doch gut umgesetzt. Vor allem haben wir uns als Ziel gesetzt, nicht mehr als zwei Treffen pro Arbeitsschritt (Pflichtenheft, Spezifikation etc.) zu brauchen, was auch gut geklappt hat, wobei manche Dinge sicher länger als andere brauchten.

Von anfänglich einem Treffen pro Woche sind wir jedoch schnell auf zwei gegangen, da wir ohne Arbeitsteilung gearbeitet haben, worauf ich im nächsten Teil noch eingehen werde. Hierfür mussten wir somit pro Woche mehr Zeit einplanen, als wir zunächst hatten.

Allgemeine Bewertung der Teamarbeit

Die Teamarbeit hat bei uns sehr gut funktioniert, da wir stets in Kontakt waren und alle gleich viel motiviert waren. Wir haben ein bisschen diskutiert, ob eine Arbeitsteilung nicht sinnvoller bzw. effizienter wäre. Letztendlich haben wir uns jedoch dagegen entschieden, da wir bevorzugt haben, dass auch jeder im Team wirklich von allem etwas mitbekommt und von jedem Aspekt sowohl des Assembler-, als auch des VHDL-Projektes eine Ahnung hatte. Dies ist uns mit dieser Methode gut gelungen.

Jeder konnte also gleich viel beitragen. Bis auf einzelne Schwierigkeiten in der Terminfindung aufgrund von unterschiedlichen Wochenplanungen der Teammitglieder, die wir aber mit etwas Kompromissfähigkeit gut bewältigt haben, war meiner Meinung nach unsere Teamarbeit sehr gut und wir hätten es im Rückblick nicht anders getan.