

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

ERA Praktikum SS 2017	1
1. Ablauf des Projekts, Probleme und Lösungen	3
2. Rückblickende Analyse für Umsetzung der Zeitplanung	4
3. Allgemeine Bewertung der Teamarbeit	5

1. Ablauf des Projekts, Probleme und Lösungen

Bei unserem ersten Treffen zu Beginn des Projekts machten wir uns einen Überblick über das zu implementierende Bauteil und wurden uns über die Rollenverteilung einig. Ergebnis dieses ersten Zusammenkommens war die Festlegung von Berzan Yildiz als Vortragenden, Thea Kramer als Verantwortliche für die Dokumentation und Florian Müller als Projektleiter.

Das darauffolgende Treffen war der Erstellung des Pflichtenhefts gewidmet. Durch die gute Vorarbeit beim parallel laufenden Assembler-Projekt, hatten wir die grobe Gliederung bereits verfügbar und konnten das Pflichtenheft für VHDL daher in einem Projekttag abschließen. Spätestens nach diesem Treffen hatten wir alle die Aufgabenstellung komplett durchdrungen und konnten uns der Erarbeitung der Spezifikation widmen.

Beim nächsten Treffen machten wir uns Gedanken über verschiedene Lösungsansätze. Im Gegensatz zum Assemblerprojekt unterschieden sich die gefundenen Ansätze jedoch nicht grundlegend, sondern eher in Detailfragen wie dem internen Aufbau der Signale. Trotzdem konnten wir bei beiden gefundenen Lösungswegen Vor und Nachteile feststellen und uns nach langem Abwägen auf einen der Lösungswege einigen. Im Laufe des Projekts gab es immer wieder Momente, in denen wir überlegten, ob nicht doch der alternative Lösungsansatz besser gewesen wäre. Trotzdem konnte sich bis zum Schluss immer wieder der gewählte Ansatz durchsetzen. Im Rahmen der Erstellung der Spezifikation, verfassten wir bereits einen Pseudocode, der syntaktisch sehr nah an echtem VHDL-Code lag und unsere Lösungsansätze verdeutlichen sollte. Für die Erstellung der Spezifikation benötigten wir insgesamt zwei Treffen.

Nachdem die Spezifikation abgeschlossen war, konnten wir uns der Implementierung des gewählten Lösungsansatzes widmen. Da wir bereits in der Spezifikationsphase einen Pseudocode geschrieben hatten, bei dem wir glaubten nur noch kleinere syntaktische Ausbesserungen vornehmen zu müssen, schätzten wir die nötige Zeit für die Implementierungsphase als überschaubar ein. Ein Problem, das wir zuvor jedoch noch nicht betrachtet hatten, war die Kompilierung und das Testen des VHDL-Codes. Da diese Themen in der vorbereitenden Vorlesung „Einführung in die Rechnerarchitektur“ nicht behandelt wurden, mussten wir uns das nötige Wissen selbst aneignen. Die vergleichsweise geringe Verbreitung von VHDL, im Gegensatz zu populäreren Programmiersprachen wie Java, machten die Suche nach Informationen schwierig. Trotzdem war schnell klar, dass GHDL ein geeigneter Simulator ist und die Signale mit GTKwave ausgewertet werden können. Das führte allerdings zum nächsten Problem, denn während die Installation von GHDL und GTKwave unter macOS mit ein paar Kommandos im Terminal schnell erledigt war, gestaltete sie sich unter Windows schwieriger als gedacht. Insgesamt benötigten wir den Großteil des ersten Treffens in dieser Phase für die Installation der benötigten Programme auf allen Rechnern.

Nachdem diese Hürde überwunden war, machten wir uns daran den Code aus der Spezifikation mithilfe von GHDL zu debuggen. Erfreulicherweise funktionierte das sehr intuitiv und es waren nur kleinere Anpassungen nötig. So konnten wir bereits nach dem nächsten Treffen einen syntaktisch korrekten VHDL-Code vorweisen. Um zu sehen, ob unsere Implementierung nicht nur syntaktisch korrekt ist, sondern auch die Aufgabenstellung erfüllt, machten wir uns an die Erstellung einer Testbench. Auch hier war zunächst Recherche nötig, aber nachdem die benötigten Informationen vorhanden waren, war die Umsetzung simpel. Als es schließlich an die Verknüpfung und Kompilierung der Testbench mit dem Hauptprogramm ging, machte sich allerdings ein neues Problem bemerkbar. Der GHDL Simulator zeigte unter Windows eine unverständliche Fehlermeldung an, während unter macOS einwandfrei kompiliert wurde. Nach langwierigen Recherchen in diversen Onlineforen stellte sich heraus, dass es in der aktuellen GHDL-Version für Windows einen Bug gab, der diese Fehlermeldung verursachte. Da die Weiterentwicklung von GHDL nur sehr langsam voranschreitet, stand nicht in Aussicht, dass der Fehler bis zum Ende der Projektarbeit behoben wäre und so entschieden wir uns, die Implementierung auf macOS fertigzustellen. Nachdem auch diese Hürde überwunden war, konnten wir die Signale der Testbench mithilfe von GTKwave visualisieren und so unser Programm verifizieren. Beim ersten Testdurchlauf wurden alle seriellen Inputwerte korrekt zu parallelen Outputwerten konvertiert und auch weitere Testdurchläufe mit verschiedenen Inputwerten führten zu der erfreulichen Erkenntnis, dass unser Programm die Aufgabenstellung erfüllte. Es waren also nur kleinere Änderungen am VHDL-Pseudocode aus der Spezifikation nötig. Trotzdem war die Implementierungsphase die zeitaufwendigste der drei Projektphase, aber auch die mit der steilsten Lernkurve.

2. Rückblickende Analyse für Umsetzung der Zeitplanung

Im Ganzen war die Zeitplanung des Projekts sehr präzise. Mit insgesamt 8 Treffen und durchschnittlich 6 Stunden pro Treffen kamen wir auf etwa 48 Stunden Arbeitszeit, statt den veranschlagten 45 Stunden. Die Zeit für Protokollierung, Dokumentation und Vorbereitung des Vortrags sind hierbei nicht eingerechnet. Für die einzelnen Projektphasen gab es allerdings teilweise Abweichungen von der Planung.

Da bei den Treffen immer alle drei Mitglieder anwesend waren, fanden die Besprechungen fließend während den anderen Phasen statt. Daher lässt sich die Zeit für Besprechungen nicht klar von den anderen Phasen trennen.

Für Aufgabenanalyse und Pflichtenheft benötigten wir etwa 6 Stunden, also etwas weniger als veranschlagt. Auch Lösungsansätze und Spezifikation konnten wir innerhalb von zwei Treffen abschließen, also ebenso etwas schneller als geplant. Hilfreich war hierbei, dass wir die Phasen des parallel laufenden Assembler Projekts immer abgeschlossen hatten, bevor wir uns dem VHDL Projekt widmeten. So hatten wir bereits einen Wissensvorsprung was die Strukturierung von Pflichtenheft und Spezifikation anging und konnten uns direkt auf den Inhalt konzentrieren.

Die Zeit die wir in den ersten beiden Projektphasen einsparen konnten, benötigten wir für die Implementierungsphase. Hier waren insgesamt vier Treffen nötig. Die meiste Zeit verbrachten wir mit Recherche und der Überwindung von technischen Hürden. Für die eigentliche Erstellung des Codes benötigten wir durch die gute Vorarbeit in der Spezifikationsphase wenig Zeit.

Zu Beginn der Projektarbeit legten wir fest, immer die Phasen des parallel laufenden Assemblerprojekts zuerst abzuschließen, um uns dann voll und ganz auf das VHDL Projekt konzentrieren zu können. Die meiste Zeit funktionierte diese Strategie sehr gut. Nur in der letzten Phase kam es durch die geplante Abwesenheit von zwei Teammitgliedern zu einem kleinen Engpass, aufgrund dessen vor dieser Abwesenheit vier Treffen in einer Woche nötig waren. Rückblickend lässt sich aber festhalten, dass auch dieser Engpass dank der hohen Motivation aller Teammitglieder sehr gut gemeistert werden konnte.

3. Allgemeine Bewertung der Teamarbeit

Die Teamarbeit hat sehr gut funktioniert. Alle Teammitglieder waren mit vollster Motivation bei der Sache und so konnten alle aufgetretenen Hürden bewältigt werden. Durch den ständigen Kontakt und die regelmäßigen Treffen, konnten alle Meilensteine fristgerecht eingehalten werden. Einzig die Terminfindung stellte aufgrund der verschiedenen Wochenpläne der Mitglieder teilweise eine Herausforderung dar. Zu Beginn des Projekts stand die Frage im Raum, ob es effektiver wäre Arbeitsteilung zu betreiben und uns nur zum Zusammentragen der Ergebnisse zu treffen. Schnell entschieden wir uns jedoch für regelmäßige Treffen, bei denen alle Teammitglieder anwesend sind und die Aufgaben gemeinsam erarbeitet werden. Hintergrund hierfür war die Intention, alle Mitglieder gleichermaßen in das Projekt zu integrieren, um Wissenslücken bei einzelnen Mitgliedern zu vermeiden und jedem die gleichen Chancen zu geben an Herausforderungen zu wachsen. Rückblickend war dies die Richtige Entscheidung, denn zu jedem Zeitpunkt wussten alle Teammitglieder über jeden Teilbereich des Projekts Bescheid, was Gruppendiskussionen und Lösungsfindung enorm vereinfachte. Durch vorausschauende Planung der Treffen und rechtzeitige Reservierung von Arbeitsräumen konnte auch die Herausforderung der Terminfindung gemeistert werden.