

Einführung in die Rechnerarchitektur Praktikum

PFLICHTENHEFT

Projekt:

MI: Komplexe Zahlen

| | |
|-------------------------|------------------|
| Projektleiter | Sandra Grujovic |
| Dokumentation | Ivan Chimeno |
| Formaler Vortrag | Martin Zinnecker |

1 Einleitung

Im Rahmen des ERA-Praktikums an der TU München sollen in Dreier-Gruppen zwei Projekte erarbeitet werden. Dieses Team besteht aus den Mitgliedern Sandra Grujovic, Ivan Chimeno und Martin Zinnecker, welche diese folgenden Projekte bearbeiten werden:

- VHDL – Am2901
- Mikroprogrammierung – Komplexe Zahlen

Dieses Pflichtenheft ist für das Mikroprogrammierung-Projekt.

2 Mikroprogrammierung – Komplexe Zahlen

2.1 Aufgabenkurzbeschreibung

Thema dieses Projekts ist die Verarbeitung der komplexen Zahlen in der mikroprogrammierbaren Maschine, im Bereich Mikroprogrammierung. Neben einer Beschreibung der komplexen Zahlen und den verbundenen Rechenregeln, werden Vor- und Nachteile des sogenannten Fixpunktformats im Vergleich zu einer Fließkommazahl besprochen. Schließlich werden MI-Befehle zur Verarbeitung und Berechnung von komplexen Zahlen als Mikroprogramm implementiert.

2.2 Ist-Zustand

Eine komplexe Zahl z ist eine Zahl, die durch zwei reelle Zahlen und eine imaginäre Zahl ausgedrückt werden kann und folgendes Format besitzt:

$$z = a + i * b \text{ mit } a, b \in \mathbb{R} \text{ und } i^2 = -1$$

Die Besonderheit der komplexen Zahl z ist die Lösbarkeit der Gleichung $x^2 + 1 = 0$ anhand ihrer imaginären Zahl.

Um komplexe Zahlen in der MI-Maschine darstellen und verarbeiten zu können, wird das sogenannte vorzeichenbehaftete (8.8) Fixpunktpunktformat verwendet. Ein Format zur Darstellung der Zahl die aus acht Vor- und Nachkommastellen konzipiert ist. Die Vor- und Nachkommastellen belegen jeweils 16 Bit im Speicher. Das heißt, insgesamt werden 32 Bits Speicher für eine komplexe Zahl in Anspruch genommen (zwei Speicherzellen). Die folgende Tabelle veranschaulicht dieses Format für eine komplexe Zahl z , die im obigen Absatz beschrieben worden ist:

| Zahl $z =$ | a | | | $+ i *$ | b | | |
|------------|----------|---|-----------|---------|----------|---|-----------|
| | Vorkomma | . | Nachkomma | | Vorkomma | . | Nachkomma |
| Bit | 16 | | | | 16 | | |

Tabelle 1: Fixpunktformat für komplexe Zahlen

Um das Projekt zu vereinfachen, ist eine Akkumulator-Architektur bereits vorhanden und besteht aus den Registern r6 und r7. Sie unterstützt verschiedene arithmetische Befehle:

- Der Akkumulator ist immer die erste beteiligte komplexe Zahl
- Die zweite beteiligte komplexe Zahl kommt immer aus dem Speicher
- Jede Berechnung speichert das Ergebnis wieder im Akkumulator
- Das Maschinenstatusregister soll sich auf den Realteil des Ergebnisses beziehen

Die Befehlssyntax des Akkumulators für die MI-Maschine hat folgendes Format:

$$[addr]: z = [addr] + i * [addr + 1] \text{ (wobei } addr \text{ ein Immediate ist)}$$

Speicheradressen können entweder von Register RA oder RB stammen, also kann die Syntax wie folgt umgeschrieben werden:

$$[RA \text{ oder } RB]: z = [RA \text{ oder } RB] + i * [RA + 1]$$

2.3 Soll-Zustand

Die zu implementierenden Befehle ergeben sich aus Folgendem:

| MI-Befehl | Beschreibung |
|-------------|--|
| cLoad [RA] | Lädt eine komplexe Zahl aus dem Speicher in den Akkumulator. |
| cStore [RB] | Kopiert den Akkumulator in den Speicher. |
| cAdd [RA] | Addiert die komplexe Zahl [RA] zum Akkumulator. |
| cSub [addr] | Subtrahiert die komplexe Zahl [addr] vom Akkumulator. |

Tests für die Befehle werden zur Verfügung gestellt.

Neben der Implementierung wird untersucht, ob sich auf dem Fixpunktformat Vor- und Nachteile gegenüber der Fließkommazahl ergeben und welcher Zahlenbereich sich mit dem Fixpunktformat darstellen lässt. Da die Befehle mit komplexen Zahlen arbeiten, werden Eigenschaften und die damit verbundenen Rechenregeln analysiert.

3 Organisation

3.1 Aufgabenverteilung

- Projektmanager – Sandra Grujovic
- Formaler Vortrag – Martin Zinnecker
- Dokumentation – Ivan Chimeno

3.2 Meilensteine

| Meilensteine | First Version | Final Version |
|-----------------|---------------|---------------|
| Pflichtenheft | 05.11.2015 | 12.11.2015 |
| Spezifikation | 26.11.2015 | 03.12.2015 |
| Implementierung | 07.01.2016 | 14.01.2016 |
| Dokumentation | 21.01.2016 | 28.01.2016 |
| Vortrag | 04.02.2016 | 11.02.2016 |

3.3 Zeitplanung

| Aufgabe | Martin Zinnecker | Ivan Chimeno | Sandra Grujovic |
|-------------------|------------------|--------------|-----------------|
| Aufgabenanalyse | 2h | 2h | 2h |
| Pflichtenheft | 3h | 3h | 3h |
| Lösungsansätze | 5h | 5h | 5h |
| Spezifikation | 5h | 5h | 5h |
| Implementierung | 11h | 11h | 11h |
| Test | 4h | 4h | 4h |
| Besprechungen | 9h | 9h | 9h |
| Vortrag | 2h | 4h | 0h |
| Vortragsfolien | 2h | 3h | 0h |
| Organisatorisches | 3h | 0h | 0h |
| Dokumentation | 0h | 0h | 7h |
| Gesamt | 45h | 45h | 45h |