

Übung 02: RISC-V Assembly

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

24. Oktober 2024



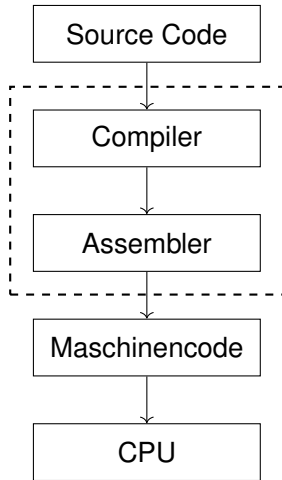
TUM Uhrenturm

Keine Garantie für die Richtigkeit der Tutorfolien.
Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

Abstraktionsebenen

- Code in einer Hochsprache (C, Java¹, ...) ist lediglich eine Abstraktion
- Compiler: Hochsprache → Assemblercode
- Assembler: Assemblercode → Maschinencode (1:1 Übersetzung)
- Maschinencode ist plattformspezifisch!
- ISA: „Bedienungsanleitung“ einer CPU

¹ im Regelfall zu Bytecode kompiliert



(Abbildung stark vereinfacht)

Reduced Instruction Set Computer

- beschränkte Menge an Instruktionen
- einfache Implementierung, schnelle Dekodierung
- komplexere Operationen benötigen mehrere Instruktionen

Beispiel: RISC-V

Complex Instruction Set Computer

- mächtiges Instruktionsset
- komplexe Implementierung, Realisierung als Mikrocode, langsame Dekodierung
- dedizierte Instruktion für beinahe jede Operation²

Beispiel: x86-64

²am Beispiel x86: VCVTTPS2UQQ, GF2P8AFFINEINVQB, MASKMOVDQU :)

- Befehlssatz in ERA: RV32IM
- 32 Register, einige davon mit spezieller Funktion
- Instruktionen auf 32 Bit begrenzt
→ Konstanten müssen zusammengebastelt werden
- Datenwortbreite: 32 Bit (4 Byte)
- Little-Endian-Architektur
- Byte-adressierbarer Speicher, maximale Hauptspeichergröße?



- Befehlssatz in ERA: RV32IM
- 32 Register, einige davon mit spezieller Funktion
- Instruktionen auf 32 Bit begrenzt
→ Konstanten müssen zusammengebastelt werden
- Datenwortbreite: 32 Bit (4 Byte)
- Little-Endian-Architektur
- Byte-adressierbarer Speicher, maximale Hauptspeichergröße?
→ 2^{32} Adressen, ca. 4.3 GB

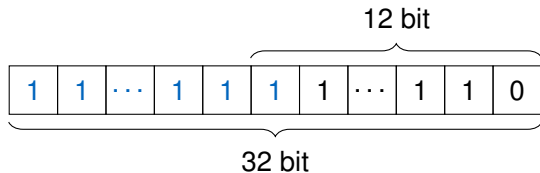
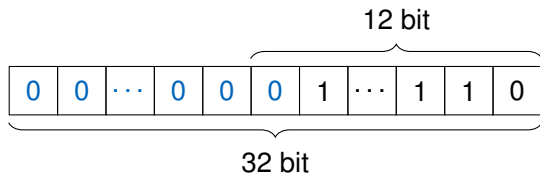


Register	ABI Name	Description	Saver ³
x0	zero	Hard-wired zero	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5–x7	t0–t2	Temporaries	Caller
x8	s0/fp	Saved register/frame pointer	Callee
x9	s1	Saved register	Callee
x10–11	a0–1	Function arguments/return values	Caller
x12–17	a2–7	Function arguments	Caller
x18–27	s2–11	Saved registers	Callee
x28–31	t3–6	Temporaries	Caller

³Erst in zwei Wochen relevant

Immediates

- als „Konstanten“ in Instruktion enkodiert
- RV32IM verwendet 12- und 20-Bit Immediates
- Datenwortbreite 32 Bit → „Erweiterung“ von Immediates notwendig
- Sign- vs. Zero-Extension
- Achtung: Immediates werden immer als signed Zahlen interpretiert!



Sign Extension

Fragen?

- Zulip: „ERA Tutorium – Mi-1600-3“ bzw. „ERA Tutorium – Fr-1500-1“
- ERA-Moodle-Kurs
- ERA-Artemis-Kurs
- Einführung Fixpunktarithmetik, Alternative
- Übersicht an RISC-V-Instruktionen

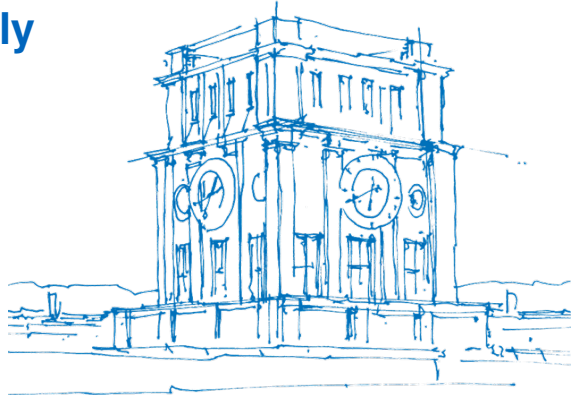
Übung 02: RISC-V Assembly

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

24. Oktober 2024



TUM Uhrenturm