

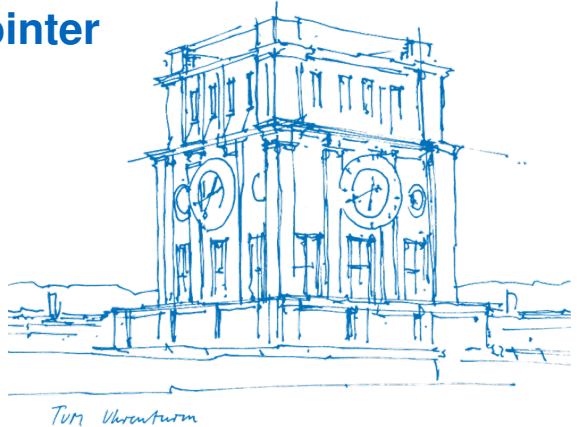
Übung 03: Sprünge und Pointer

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

1. November 2024



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

Branch-Befehle

Rücksprungadresse wird nicht gesichert,
Sprungbedingung muss erfüllt sein

- `beq: $rs1 = rs2$`
- `bne: $rs1 \neq rs2$`
- `blt(u): $rs1 < rs2$`
- `bgt(u): $rs1 > rs2$`

Jump-Befehle

Schreiben Rücksprungadresse in ra oder
angegebenes Register, springen immer

- `jal label`
- `jalr rd, offset(rs)`
- `j label` Achtung, überschreibt ra nicht!

lade 32 Bit an der Adresse $a0 + 0$ Bytes Offset in das Register $t0$:

```
lw t0, 0(a0)
```

lade 8 Bit an der Adresse $a2 - 4$ Bytes Offset in das Register $t1$:

```
lb t1, -4(a2)
```

speichere den gesamten Inhalt des Registers $t2$ an die Adresse $a1 + 16$ Bytes Offset:

```
sw t2, 16(a1)
```

<MINTED>

<MINTED>

ro: read-only, **rw**: les- und schreibbar

Fragen?

(Die ZÜ-Folien sind sehr gut, schaut euch die an)

- „H03 — Palindromerkennung“ bis 10.11.2024 23:59 Uhr
- Verwendung von Speicheroperationen, Unterprogrammaufrufe
- Pseudoinstruktion `tail` für tailcalls, verwendet aktuellen Stackframe wieder

- [Zulip: „ERA Tutorium - Do-1600-1“ bzw. „ERA Tutorium - Fr-1500-2“](#)
- [RISC-V-Spezifikation](#)
- [ERA-Moodle-Kurs](#)
- [ERA-Artemis-Kurs](#)
- [Übersicht an RISC-V-Instruktionen](#)
- [GNU as directives](#)

Übung 03: Sprünge und Pointer

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

1. November 2024

