

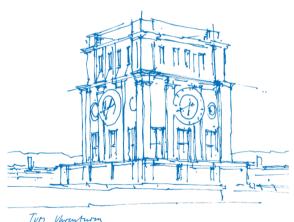
Übung 03: Sprünge und Pointer

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!

Sprungbefehle



Branch-Befehle

Rücksprungadresse wird nicht gesichert, **Sprungbedingung** muss erfüllt sein

- \blacksquare 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!

Speicherzugriffe



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:

1b t1, -4(a2)

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

sw t2, 16(a1)

Sections und Direktiven



```
# compile-time Konstante
.egu NUM. 2748
# ro + init Daten
.rodata
 f: .word 2
# r/w + init. Daten
.org 0x400
data
  arr: .byte 4. 3. 2. 1
  string1: .ascii "asdf"
  string2: .asciz "asdf"
```

```
\# r/w + uninit. Daten
hss
 a: .space 16
# definiere globales Einstiegslabel
.globl _start
.org 0x200 # section startet an
\rightarrow Adresse 0x200
.text
start:
  la a0, arr
   lbu a1. 0(a0)
```

ro: read-only, r/w: les- und schreibbar



Fragen?

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

Artemis-Hausaufgaben



- "H03 Palindromerkennung" bis 10.11.2024 23:59 Uhr
- Verwendung von Speicheroperationen, Unterprogrammaufrufe

Links



- 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

