

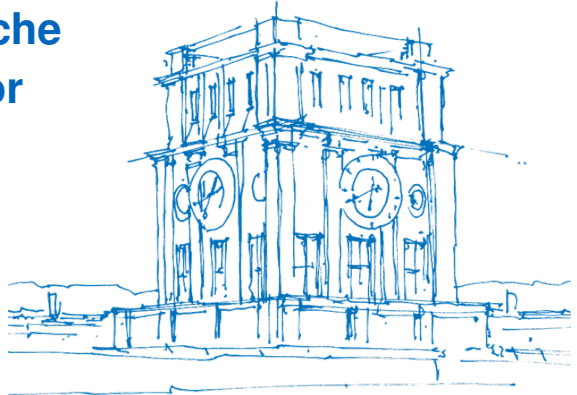
Übung 08: Maschinensprache und Single-Cycle-Prozessor

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

5. Dezember 2025



TUM Uhrenturm

t1p.de/era2526



home.in.tum.de/~ladu/



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

| | | |
|----|--|----------------|
| R | Register-Register-Operationen | add, sub, sll |
| I | Short Immediates (12 Bit) und Ladebefehle | jalr, lw, ori |
| S | Speicherbefehle | sw, sh |
| B | Branches (bedingte Sprünge) | beq, blt, bgtu |
| U | Long Immediates (20 Bit) | lui, auipc |
| J | Jumps (unbedingte Sprünge) mit Long Immediate | jal |
| R4 | Floating-Point-Operationen, für ERA nicht relevant | |

¹NB: Die Befehle für die einzelnen Typen sind nur auszugsweise angegeben.

| 31:25 | | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 | |
|---------------------------------|--------|--------|--------|-----------------------|--------|--------|---------|
| funct7 | | rs2 | rs1 | funct3 | rd | op | R-Type |
| imm _{11:0} | | | rs1 | funct3 | rd | op | I-Type |
| imm _{11:5} | rs2 | rs1 | funct3 | imm _{4:0} | op | | S-Type |
| imm _{12,10:5} | rs2 | rs1 | funct3 | imm _{4:1,11} | op | | B-Type |
| imm _{31:12} | | | | | rd | op | U-Type |
| imm _{20,10:1,11,19:12} | | | | | rd | op | J-Type |
| fs3 | funct2 | fs2 | fs1 | funct3 | fd | op | R4-Type |
| 5 bits | 2 bits | 5 bits | 5 bits | 3 bits | 5 bits | 7 bits | |

(Quelle: Vorlesungsmaterialien ERA)

```
xor t2, t1, t0
```

1. Instruktionstyp feststellen

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen → R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen → R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

Assemblierung: Beispiel

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

xor t2, t1, t0

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

funct7 rs2 rs1 funct3 rd op

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

funct7 rs2 rs1 funct3 rd op

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

0000000 rs2 rs1 100 rd 0110011

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

0000000 rs2 rs1 100 rd 0110011

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

0000000 rs2 rs1 100 rd 0110011

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

0000000 00101 00110 100 00111 0110011

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

0000000 00101 00110 100 00111 0110011

Assemblierung: Beispiel

xor t2, t1, t0

1. Instruktionstyp feststellen \rightarrow R
2. Dazugehöriges Layout in Tabelle finden
3. Instruktion in Tabelle finden
4. Instruktionsspezifische Werte ablesen
5. Registermapping:
(zero \mapsto x0, ..., t6 \mapsto x31)
6. Binärzahl zusammenbauen

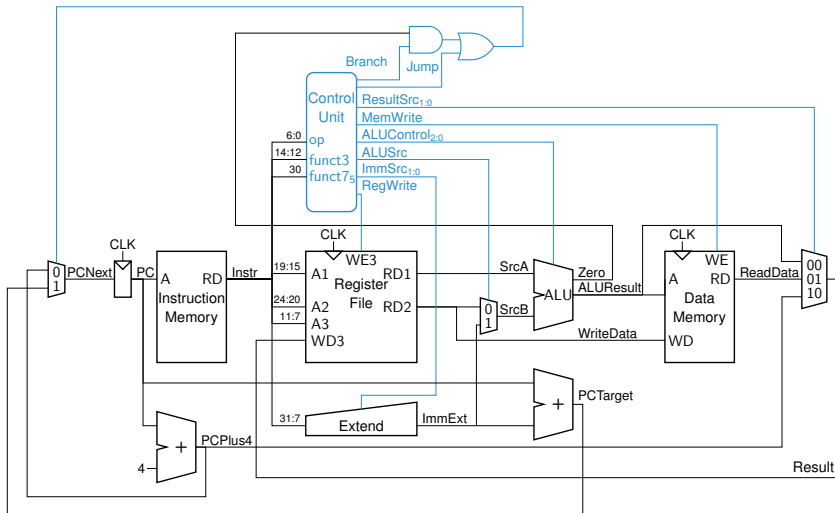
| 31:25 | 24:20 | 19:15 | 14:12 | 11:7 | 6:0 |
|--------|-------|-------|--------|------|-----|
| funct7 | rs2 | rs1 | funct3 | rd | op |

| op | funct3 | funct7 | Type | Instruction |
|--------------|--------|---------|------|------------------|
| 0110011 (51) | 100 | 0000000 | R | xor rd, rs1, rs2 |

t2 \mapsto x7, t1 \mapsto x6, t0 \mapsto x5

0b 0000 0000 0101 0011 0100 0011 1011 0011 =
0x005343B3

RISC-V Single-Cycle-Prozessor



Fragen?

- Zulip: „ERA Tutorium – Mi-1600-3“ bzw. „ERA Tutorium – Fr-1500-1“
- ERA-Moodle-Kurs
- ERA-Artemis-Kurs
- Prozessor-Assets (kein offizielles Material!)
- RISC-V Assembler

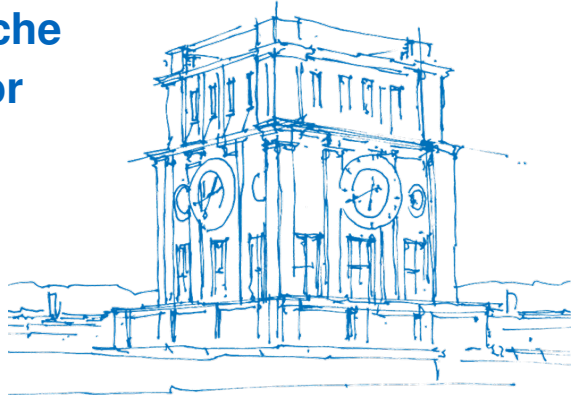
Übung 08: Maschinensprache und Single-Cycle-Prozessor

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

5. Dezember 2025



TUM Uhrenturm