

Übung 03: Unterprogramme und Speicherzugriffe

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

4. November 2023



TUM Uhrenturm

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

Sprünge und Unterprogramme

- if, while, for → umgesetzt durch Sprünge
- jumps: unbedingt, branches: bedingt
- in x86 über Flags-Register, in RISC-V direkt im Befehl eingebaut
- Unterprogrammaufrufe: Sicherung, Parameter, Sprung + Rückkehr

Caller- und Callee-saved Register

Register	ABI Name	Description	Saver
x0	zero	Hard-wired zero	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5–7	t0–2	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
f0–7	ft0–7	FP temporaries	Caller
f8–9	fs0–1	FP saved registers	Callee
f10–11	fa0–1	FP arguments/return values	Caller
f12–17	fa2–7	FP arguments	Caller
f18–27	fs2–11	FP saved registers	Callee
f28–31	ft8–11	FP temporaries	Caller

Abbildung 1 Übersicht über die RISC-V-Register

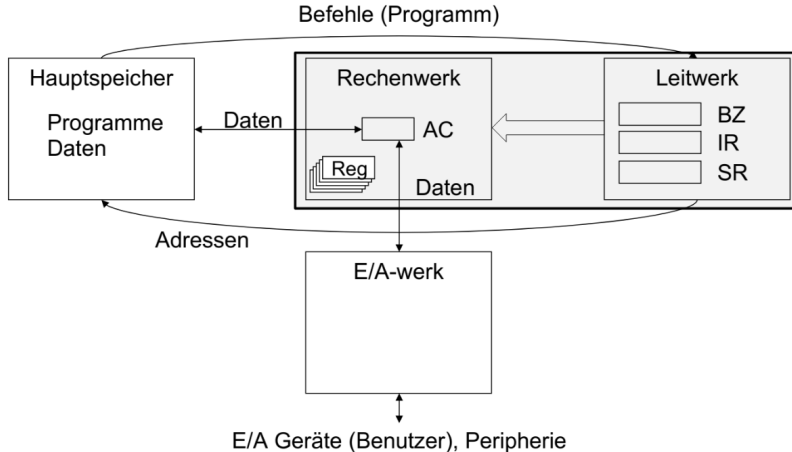


Abbildung 2 Schematische Darstellung der von-Neumann-Architektur

- Hauptspeicher = RAM, nicht Festplatten etc.
- Laden/speichern mittels lb/sb/lw/sw
- Byte-Befehle laden in die unteren 8 Bit des Registers
- Register als Basisadresse + Offset

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)
```

Fragen?

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

Quiz

- H03 - Palindrom bis 12.11.2023 23:59 Uhr
- Speicheroperationen
- lasst euch nicht vom Label 'hard' abschrecken, Prinzip ist sehr simpel

- Zulip: „ERA Tutorium - Mi-1600-MI4“ bzw. „ERA Tutorium - Fr-1100-MW2“
- RISC-V-Spezifikation
- Von-Neumann-Architektur
- übersichtlichere Instruktionsliste

Übung 03: Unterprogramme und Speicherzugriffe

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

4. November 2023



TUM Uhrenturm