ICS
Institute for Complex Systems

Bachelorinfoabend 2026

Univ.-Prof. Dr.-Ing. Daniel Große
jku.at/ics
Email: daniel.grosse@jku.at

JOHANNES KEPLER UNIVERSITY LINZ

news

CHIPGIGA

Poker

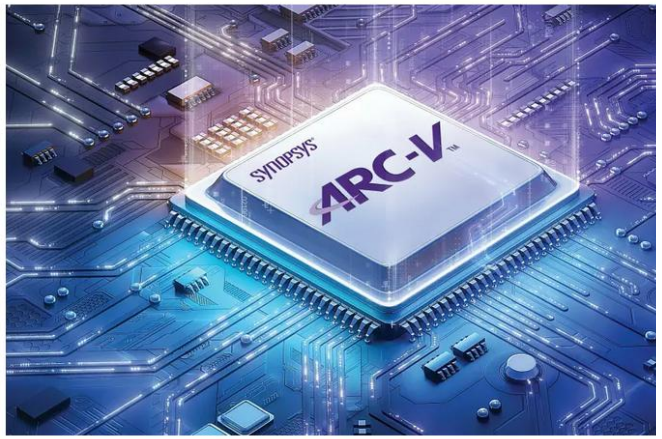
Ein geplant
ganzen We
dessen Tec
dem japani
Arm nun in

8. Septembe

RISC-V-Kerne in gängigem Chip-Entwicklerwerkzeug von Synopsys integ

Synopsys, der große Anbieter von Software für Chipdesigner (EDA-Too
nun auch RISC-V-Kerne mit 32 oder 64 Bit zum SoC-Einbau.

Lesezeit: 2 Min. In Pocket speichern



Synopsys ARC-V: Eingebettete RISC-V-Kerne (Bild: Synopsys)

08.11.2023 11:55 Uhr | c't Magazin

Von Christof Windeck

Sollte die Übernahme von Arm durch Nvidia tatsächlich A
auf die weltweite Rechnerproduktion haben, könnte das
längerfristig – dazu führen, dass sich Hersteller nach Alter
den Arm-Designs umsehen. Einen Kandidaten – bisher gro
Außenseiter – gibt es dafür schon. Mit RISC-V existiert ein
Konkurrenzstandard mit einem enormen Vorteil: Die
Rechnerarchitektur ist offen verfügbar – das heißt, für die
in Chips fällt an sich keine Lizenzgebühr an.

Aktuell | Prozessoren c't 16/2024 S. 36

Bit-Rauschen

Aufwind für die offene Pro

**Der KI-Goldrausch tobt weiter. Grö
ein Stück vom Kuchen. Intel feiert
superschnell bearbeiten.**

Von Christof Windeck

RISC-V strebt aufwärts

Sie sind da, obwohl man sie nicht sieht: I
Im laufenden Jahr 2024 kommen laut de
hungsweise SoCs mit RISC-V-Technik zu
32-Bit-Kernchen für eingebettete Mikroc
diatek oder in der ESP32-C-Serie von Esp

Auf dem RISC-V Europe Summit 2024 in
ein Gespräch mit dem emeritierten Berke
wartet erhebliches Wachstum bei 64-Bit-
Server, Autos und Unterhaltungselektron
chinesischen Markt. Die spanische Firma
eingebauten KI-Rechenbeschleunigern. D
ne dass diese erst aufwendig mit DMA-Z
müssen.

heise online > Technik > Mikrocontroller > RISC-V fürs Auto: Infineon kündigt Mikrocontroller mit neuen Reche

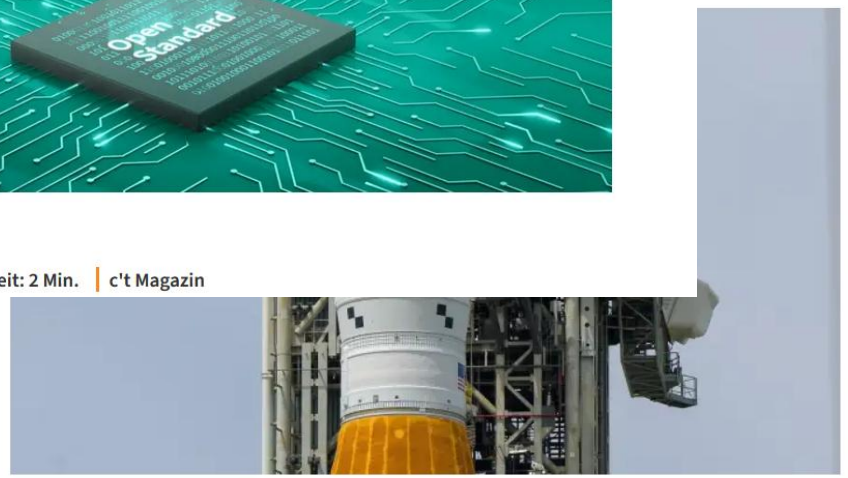
RISC-V fürs Auto: Infineon kündigt Mikrocontroller mit neuen Rechenkernen an

Der Marktführer bei Mikrocontrollern für Fahrzeuge setzt künftig auch auf die offengelegte Befehlssatzarchitektur RISC-V, die mit ARM-Kernen konkurriert.

1

(Bild: Infineon)

06.03.2025, 13:04 Uhr Lesezeit: 2 Min. | c't Magazin



Die NASA setzt beim High-Performance Spaceflight Computing (HPSC) künftig auf RISC-V. Ein europäischer Nanosat mit RISC-V-CPU umkreist bereits die Erde.

tom's HARDWARE US Edition

Reviews Best Picks

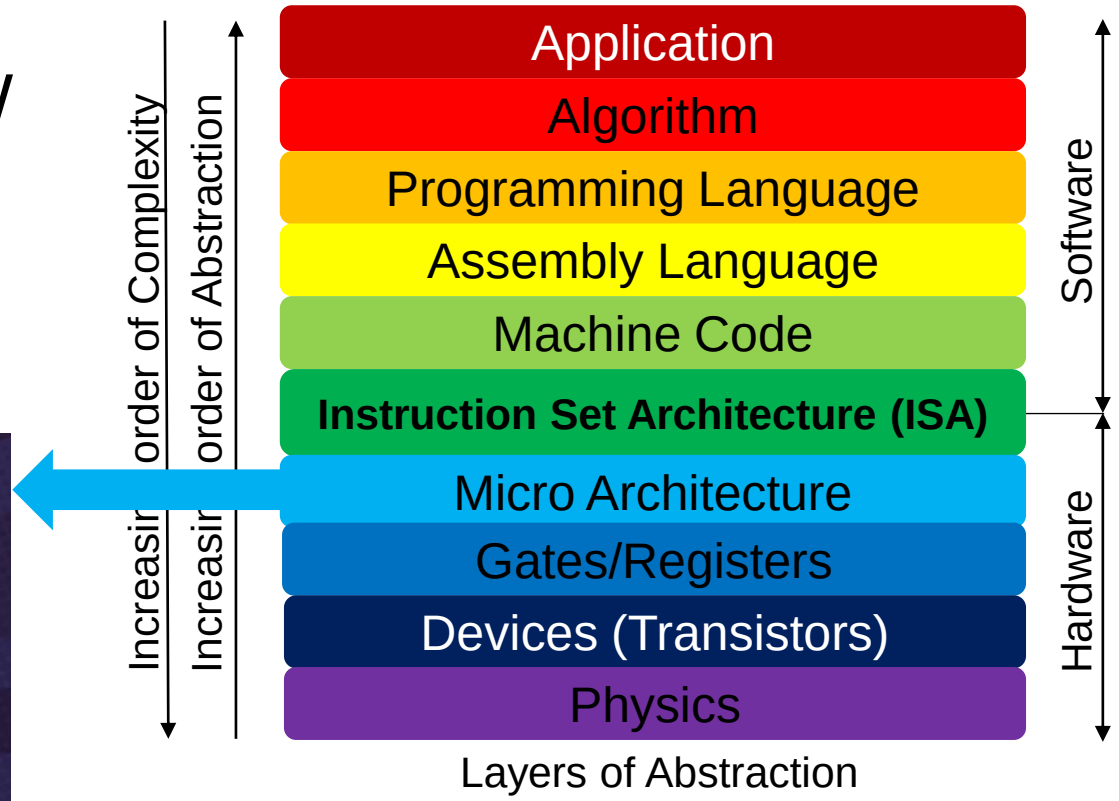
Intel Offers $2 Billion for RISC-V Chip Startup SiFive: Bloomberg

By Paul Alcorn June 10, 2021

Instruction Set Architecture (ISA)

- Instruction Set Architecture (ISA)

  is the interface between HW and SW
  - Instructions (encoding & semantics)
  - Number and types of registers
  - Memory access, addressing modes



Modular Design

Intel x86, ARM
Incremental instruction set

RISC-V
Composable extensions

Disruptive **Technology**

Barriers	Legacy ISA	RISC-V ISA
Complexity	1500+ base instructions Incremental ISA	47 base instructions Modular ISA
Design freedom	$$$ – Limited	Free – Unlimited
License and Royalty fees	$$$	Free
Design ecosystem	Moderate	Growing rapidly. Numerous extensions and cores. More design companies on RISC-V than any other architecture
Software ecosystem	Extensive	Growing rapidly. Easy to compile for RISC-V
Specification	180 hours to read	6 hours to read

RISC-V®

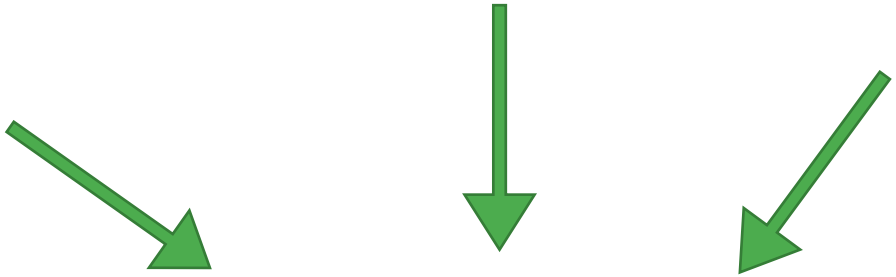
RISC-V is an open standard

Bachelor Thesis

I want to implement an AI accelerator!

I want to work on SW side or programming languages!

I want to implement my own processor!

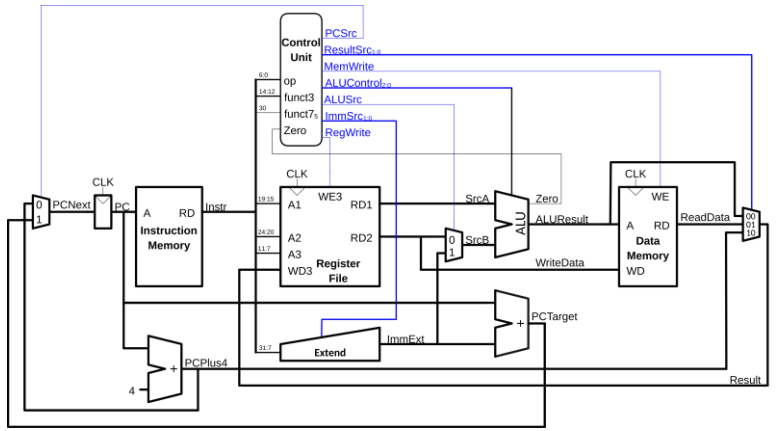


Contact us

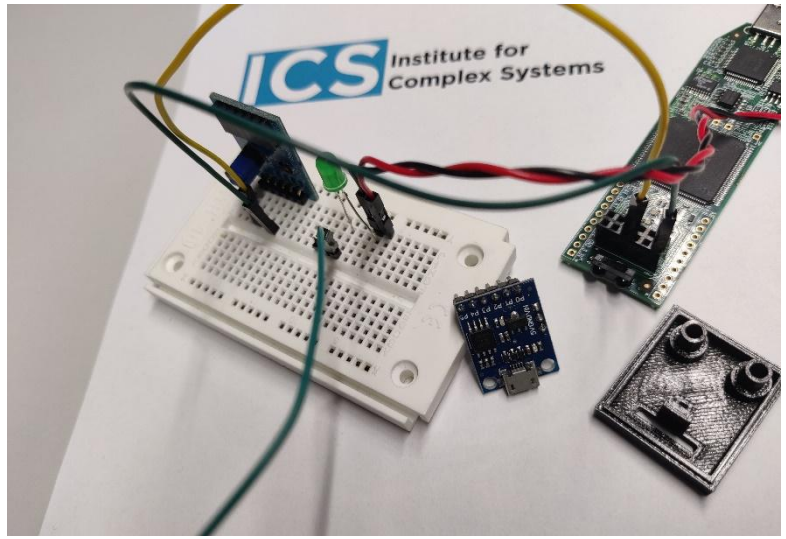
- **Together** we will work out the topic of your thesis
- Depending on **your** interests
- Already have an idea? Great!
- We support you at every stage of your thesis

A very cool result
And a new title for your collection

Research Areas at our Institute



- Modelling & Implementation
  - RISC-V virtual prototypes
  - RISC-V cores (VHDL, Verilog)
  - SUBLEQ Processor (processor in 1 instruction)
  - Accelerators (RISC-V Vector extension, NNs, Approximate Computing)
  - …
- Electronic Design Automation: e.g. verification
  - Property Checking / Equivalence Checking
  - Waveform Analysis Language (WAL)
  - Fuzzing
  - Metamorphic Testing
  - Symbolic execution of SW interacting with HW
  - Symbolic Computer Algebra (SCA) for hard arithmetic (e.g. multipliers) at gate level



→ **More Details:** https://ics.jku.at/research/

Editor | Processor | Waveform

```
1  # Init x1=1; x2=2
2  addi x1, zero, 1
3  addi x2, zero, 2
4
5  # b'01 xor b'10 = add 1 + 2
6  # Data conflict (RAW):
7  #    Reads x1 but addi x1 not finished
8  #    Reads x2 but addi x2 not finished
9  # Solution:
10 #    1 x nop does not solve conflict
11 #    2 x nop only solves addi x1
12 #    3 x nop solves both
13 add x4, x1, x2
14
```

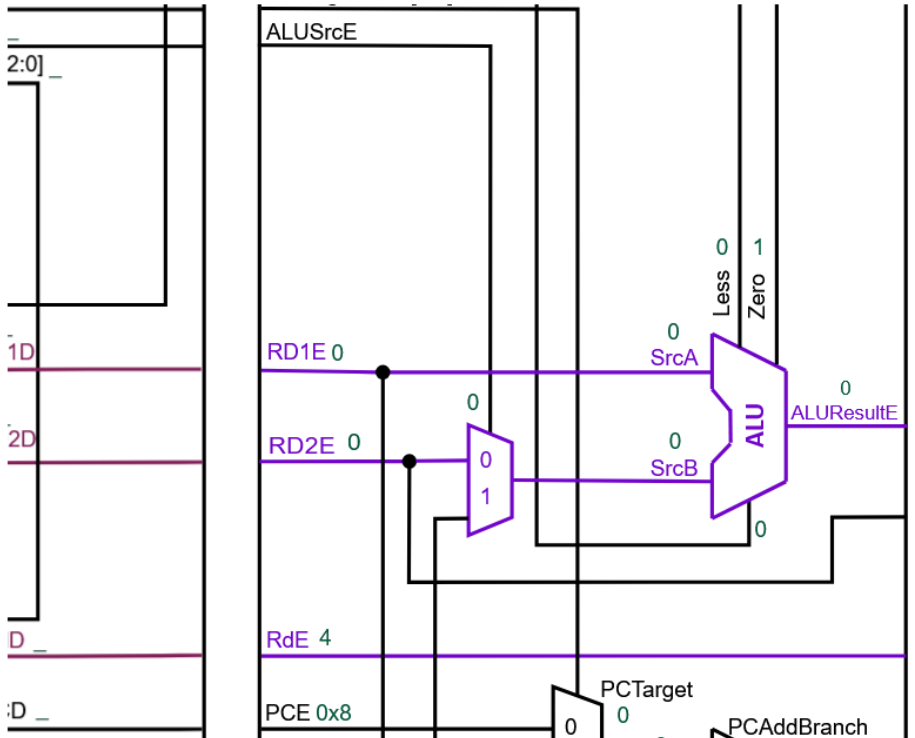
Simulation options

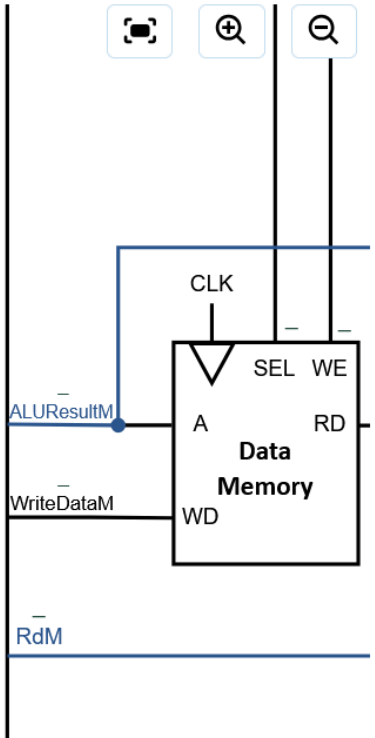
- ○ ICS-EDU-RV32I-SC (new version)
- ○ ICS-EDU-RV32I-SC
- ○ ICS-NEX-RV32I-SC
- ○ ICS-EDU-RV32I-MC
- ○ ICS-NEX-RV32I-MC
- ◉ ICS-NEX-RV32I-PL
- ○ ICS-NEX-RV32I-PLH

Simulate now

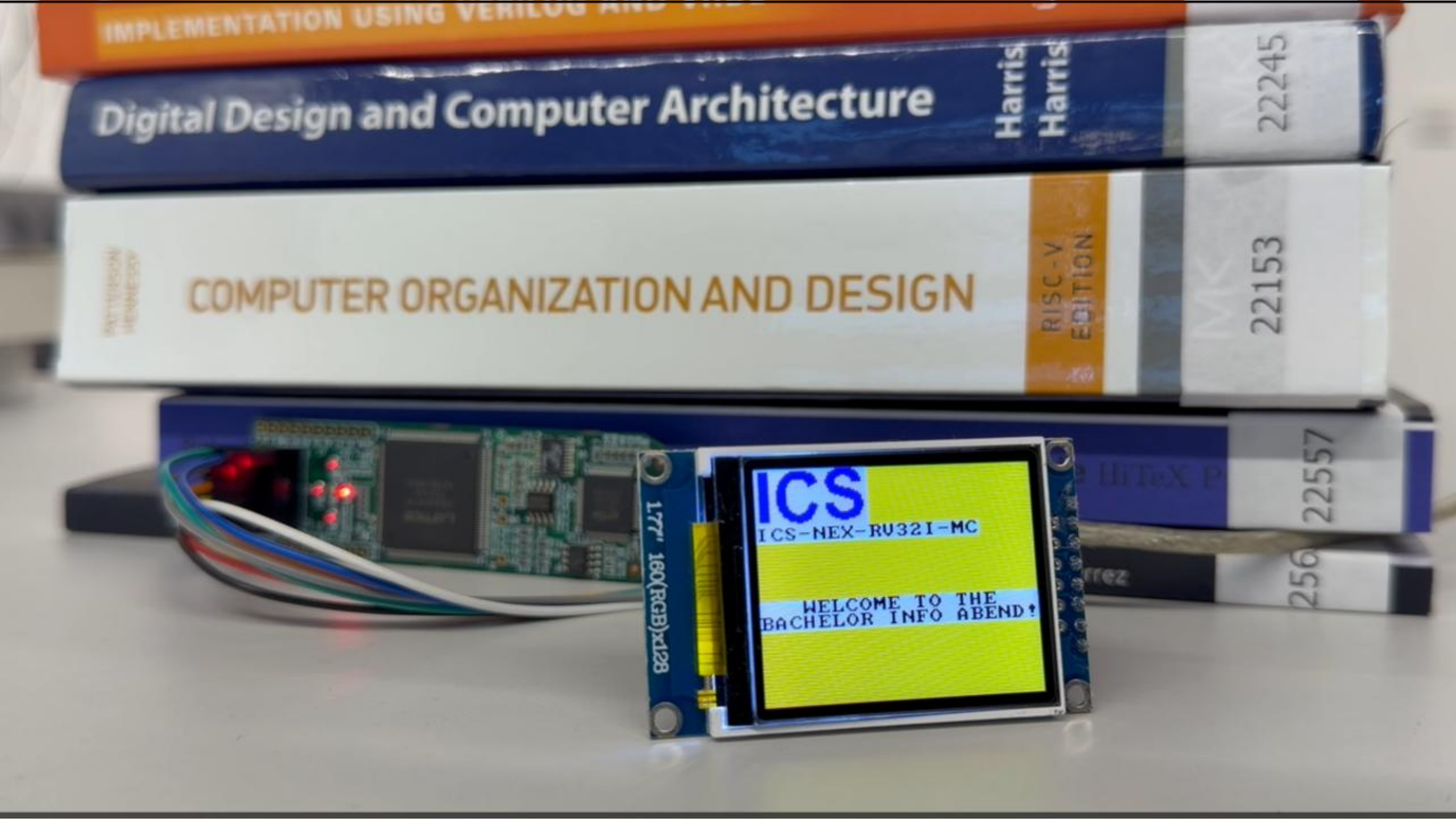
SonicRV | Examples | Editor | Processor | Waveform

PC	fetch	decode	execute	memory	writeback
0	addi x1, x0, 1				
4	addi x2, x0, 2	addi x1, x0, 1			
8	add x4, x1, x2	addi x2, x0, 2	addi x1, x0, 1		
C	bne x3, x4, .+12	add x4, x1, x2	addi x2, x0, 2	addi x1, x0, 1	
10	addi x5, x0, 1	bne x3, x4, .+12	add x4, x1, x2	addi x2, x0, 2	addi x1, x0, 1
14	jal x0, .+8	addi x5, x0, 1	bne x3, x4, .+12	add x4, x1, x2	addi x2, x0, 2
18	addi x5, x0, 0	jal x0, .+8	addi x5, x0, 1	bne x3, x4, .+12	add x4, x1, x2
1C	addi x6, x0, 1	addi x5, x0, 0	jal x0, .+8	addi x5, x0, 1	bne x3, x4, .+12
1C	addi x6, x0, 1	addi x6, x0, 1	addi x5, x0, 0	jal x0, .+8	addi x5, x0, 1
20		addi x6, x0, 1	addi x6, x0, 1	addi x5, x0, 0	jal x0, .+8
24			addi x6, x0, 1	addi x6, x0, 1	addi x5, x0, 0
28				addi x6, x0, 1	addi x6, x0, 1
2C					addi x6, x0, 1

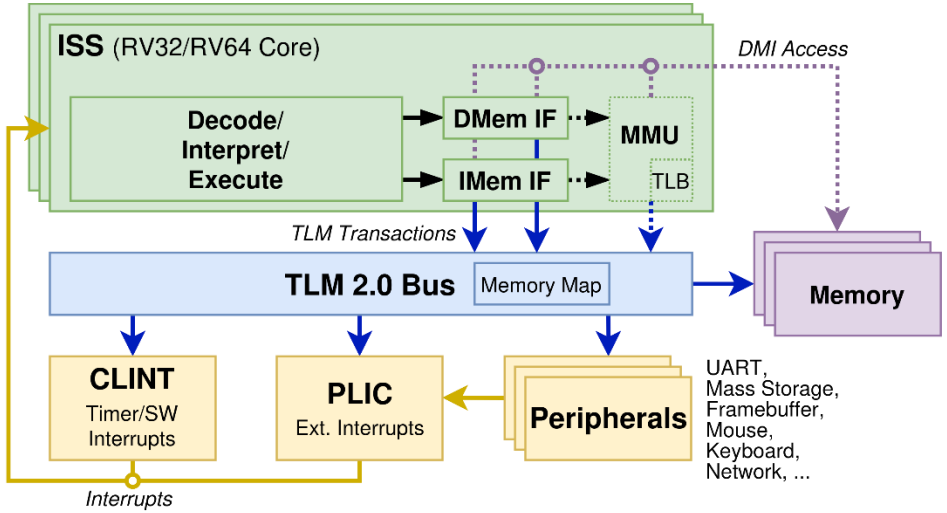




Register	Value
x1	0
x2	0
x3	0
x4	0
x5	0
x6	0

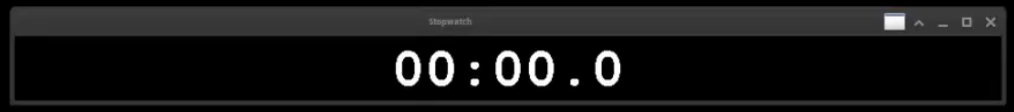

IMPLEMENTATION USING VERILOG AND
Digital Design and Computer Architecture
Harris
Harris
22245
COMPUTER ORGANIZATION AND DESIGN
RISC-V
EDITION
22153
22557
ICS
ICS-NEX-RV32I-MC
WELCOME TO THE
BACHELOR INFO ABEND!
1.77" 160(RGB)x128

RISC-V VP++ Overview



- Open source on GitHub
  - https://github.com/ics-jku/riscv-vp-plusplus
- Key features:
  - SystemC/C++ TLM-2.0
  - 32 & 64 bit cores (RV32GCV + SUN, RV64GCV + SUN)
  - Bare metal/Small operating systems configurations, including:
    - SiFive HiFive1 - FE310
    - GD32VF103VBT6 microcontroller (Nuclei N205) including UI
  - Linux RV32 and RV64, single and quad-core VPs (SiFive FE540)
  - Full integration of GUI-VP, which enables simulation of interactive graphical Linux applications
  - Support for *RISC-V "V" Vector Extension* (RVV) version 1.0
  - Support for *Capability Hardware Enhanced RISC Instructions* (CHERI)
  - Based on RISC-V VP introduced in 2018

Linux Boot


Stopwatch
00:00.0


Terminal - ame@pulsar: /srv/ext/VP/GUI-VP_Kit
ame@pulsar:/srv/ext/VP/GUI-VP_Kit$ make run_rv64_sc

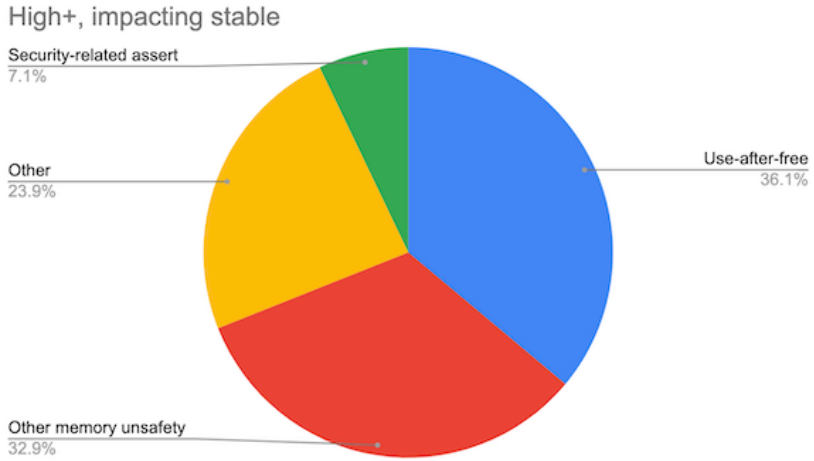
Linux PrBoom Demo Run


```
 found /usr/share/games/doom/doom1.wad
IWAD found: /usr/share/games/doom/doom1.wad
PrBoom (built Feb  3 2025), playing: DOOM Shareware
PrBoom is released under the GNU General Public license v2.0.
You are welcome to redistribute it under certain conditions.
It comes with ABSOLUTELY NO WARRANTY. See the file COPYING for details.
V_Init: allocate screens.
 found /usr/share/games/doom/prboom.wad
D_InitNetGame: Checking for network game.
W_Init: Init WADfiles.
 adding /usr/share/games/doom/doom1.wad
 adding /usr/share/games/doom/prboom.wad
W_InitCache

M_Init: Init miscellaneous info.
R_Init: Init DOOM refresh daemon -
R_LoadTrigTables: Endianness...ok.
R_InitData: Textures Flats Sprites Tranmap build [........]
R_Init: R_InitPlanes R_InitLightTables R_InitSkyMap R_InitTranslationsTables R_InitPatches
P_Init: Init Playloop state.
I_Init: Setting up machine state.
I_InitSound: couldn't open audio with desired format
S_Init: Setting up sound.
S_Init: default sfx volume 8
HU_Init: Setting up heads up display.
I_InitGraphics: 640x480
I_UpdateVideoMode: 640x480 (fullscreen)
V_InitMode: using 8 bit video mode
I_SetRes: Using resolution 640x480
I_UpdateVideoMode: 0xe0000000, SDL buffer, direct access
ST_Init: Init status bar.
```

~12 FPS @ 640x480
~12 Mio per Frame
~144 MIPS
(each pixel is a TLM transaction)

HW and Security: CHERI



- We are haunted by memory safety issues
- Enforcing memory safety is a non-trivial problem
- Classical answer:
  - The programmer forgot to check the bounds of the data structure being read
  - Fix the vulnerability in hindsight – one-line fix:

    if (1+2+payload+16 **>** s->s3->rrec.length) **return** 0;
- How to do better?
  - Preserve bounds information during compilation
  - Use hardware (CHERI processor) to dynamically check bounds with little overhead and guarantee pointer integrity & provenance
- ICS: CHERI in RISC-V VP …

BA Beispiele (Auswahl)

Web: https://ics.jku.at/teaching/theses/

- “Design Space Exploration of Sequential RISC-V Microarchitectures” von Jonas Karg (Februar 2026)
- “Integration and Visualization of a RISC-V Multi-Cycle Core in SonicRV” von Jan Rendel (Januar 2026)
- “A Web-Based Instruction Decoder with Semantic Explanations for RISC-V” von Tobias Kathan (September 2025)
- “Design and Verification of Interrupt Support in a Single-Cycle RISC-V Processor” von Thomas Schrott (Juni 2025)
- “Visualization of Transaction-Level Model Simulations in Surfer” von Christian Dattinger (Februar 2025).
- “ProtoLens: Visualizing Transaction Dynamics in Virtual Prototypes” von Jonas Reichhardt (Februar 2025).
- “RISC-V Multi-Cycle Processor Implementation in VHDL” von Oliver Trost (Oktober 2024).
- “Static Polymorphism in Spade using Traits” von Fabian Bleck, Alexander Pichler und Rene Wimmer (August 2024).
- “First SERV: Towards Device Localization using BLE” von Paul Blume (Mai 2024).
- “From Model to Metal – The SUBLEQ microcoded microprocessor Goldcrest on the ICEStick1k” von Felix Roithmayr (Dezember 2023).
- “Enhancing a Virtual Prototype with RISC-V Vectoring Extension: Implementation and Verification” von Moritz Stockinger (September 2023).

Contact us!

- **Univ.-Prof. Dr. Daniel Große**
- **Science Park 4, 3rd floor**
- **daniel.grosse@jku.at**
- **jku.at/ics**

Bachelor- and Masterthesis
https://ics.jku.at/teaching/theses/

also: Seminars / Projects

Target: Mastering the constantly growing complexity of hardware/software systems

Topics:
- Virtual Prototypes (VPs) for HW/SW systems (System Level)
- HW designs in Verilog/VHDL at the Register Transfer Level (RTL), down to the gate-level

Primary research areas:
- Verification, debugging, hardware security, and synthesis
- Problems in Electronic Design Automation (EDA)

We heavily use and contribute to open-source (see e.g. RISC-V VP++, Waveform Analysis Language WAL, Surfer, …)

ICS
Institute for Complex Systems

Bachelorinfoabend 2026

Univ.-Prof. Dr.-Ing. Daniel Große
jku.at/ics
Email: daniel.grosse@jku.at

JOHANNES KEPLER UNIVERSITY LINZ