

五、目标代码生成

(13. RISC-V 概述)

魏恒峰

hfwei@nju.edu.cn

2024 年 05 月 24 日



RISC: Reduced Instruction Set Computer (精简指令集计算机)



V: Five

CISC: Complex Instruction Set Computer (复杂指令集计算机)

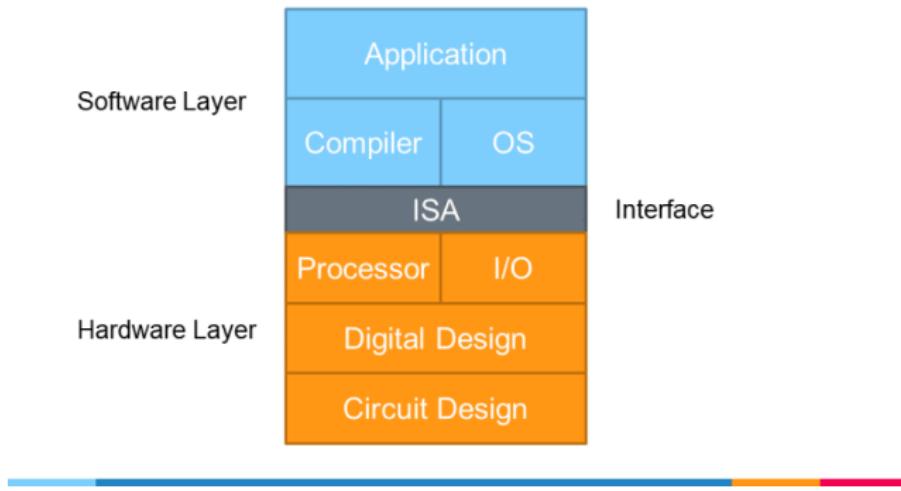


RISC: Each instruction performs only one function.

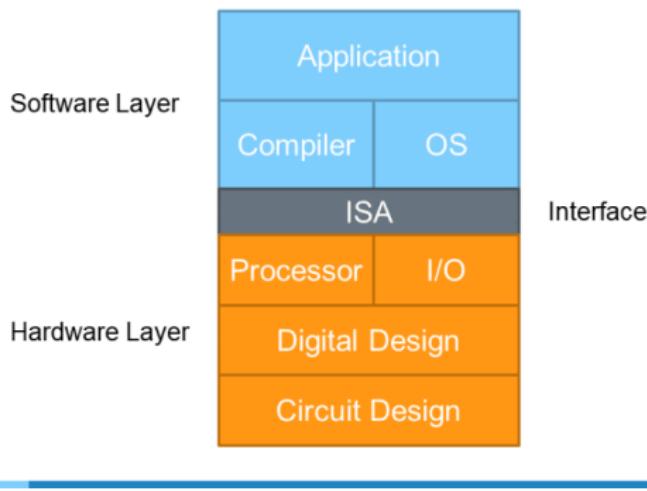
CISC: SUBL val, %eax ; %eax <- %eax - val



ISA (Instruction Set Architecture) as the Software/Hardware Interface



ISA (Instruction Set Architecture) as the Software/Hardware Interface



In general, an ISA defines the supported **instructions**, **data types**, **registers**, the hardware support for managing **main memory**, fundamental features (such as the **memory consistency**, **addressing modes**, **virtual memory**), and the **input/output** model of a family of implementations of the ISA.

Krste Asanović at the University of California, Berkeley, had a research requirement for an open-source computer system, and in 2010 he decided to develop and publish one in a "short, three-month project over the summer" with several of his graduate students. The plan was to aid both academic and industrial users.^[12] David Patterson at Berkeley joined the collaboration as he was the originator of the Berkeley RISC,^[18] and the RISC-V is the eponymous fifth generation of his long series of cooperative RISC-based research projects at the University of California, Berkeley (RISC-I and RISC-II published in 1981 by Patterson, who refers^[21] to the SOAR architecture^[22] from 1984 as "RISC-III" and the SPUR architecture^[23] from 1988 as "RISC-IV"). At this stage, students provided initial software, simulations, and CPU designs.^[24]

2017 ACM Turing Award



“Hennessy and Patterson created a systematic and quantitative approach to designing faster, lower power, and **reduced instruction set computer (RISC)** microprocessors.”





[About RISC-V](#)[Membership](#)[RISC-V Exchange](#)[Technical](#)[News & Events](#)[Community](#)

Announcements

First-ever RISC-V Summit Europe Will Demonstrate Technical and Commercial Momentum Across Industries

The Barcelona RISC-V Summit from June 5-9 will focus on industries such as Automotive, High-Performance Compute/Data Center, and Security; Call for Submissions and Sponsorships are now open.

[READ MORE](#)

Join RISC-V International



RISC-V International comprises a large member organization building the first open, collaborative community of software and hardware innovators powering innovation at the edge forward. Through various events and workshops, RISC-V International is changing the way the industry works together and collaborates – creating a new kind of open hardware and software ecosystem. Become a member today and help pioneer the industry's future!

Premier Members



成为资本 CHENGING CAPITAL





自助餐 (大而全, 增量式; x86)

FAST FOOD MENU

Main Course

Cheese Burger	\$3
Cheese Sandwich	\$2
Chicken Burger	\$3
Spicy Chicken	\$3
Spicy Burger	\$3

Appetizers

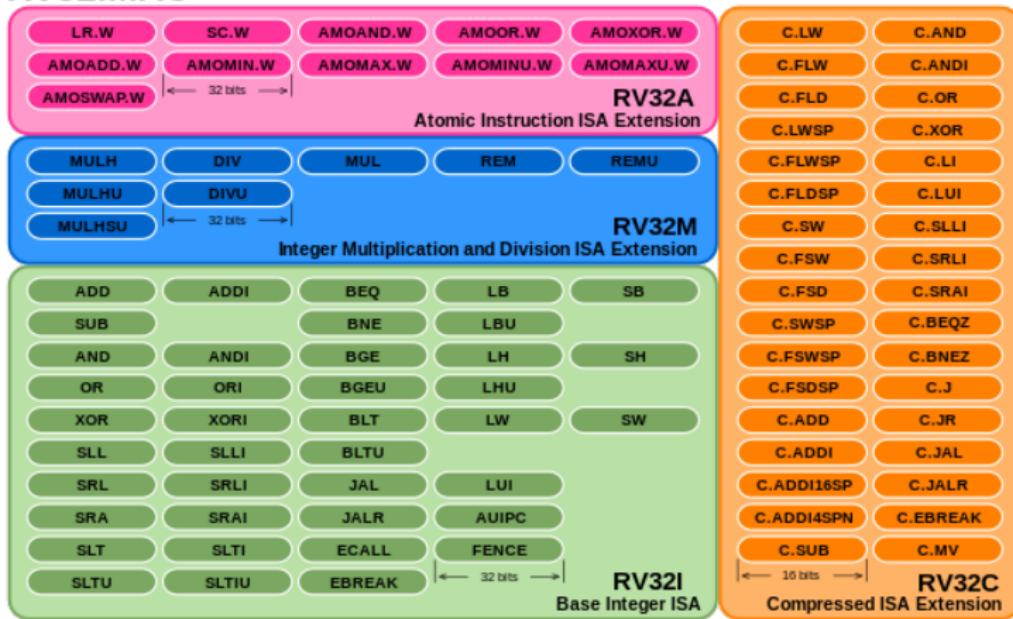
French Fries	\$1
Chicken Nugget	\$1
Ice Cream	\$1
Cheese Cake	\$1

Drinks

Milkshake	\$1
Ice Tea	\$1
Orange Juice	\$1
Lemon Tea	\$1

菜单 (模块式; RISC-V)

RV32IMAC

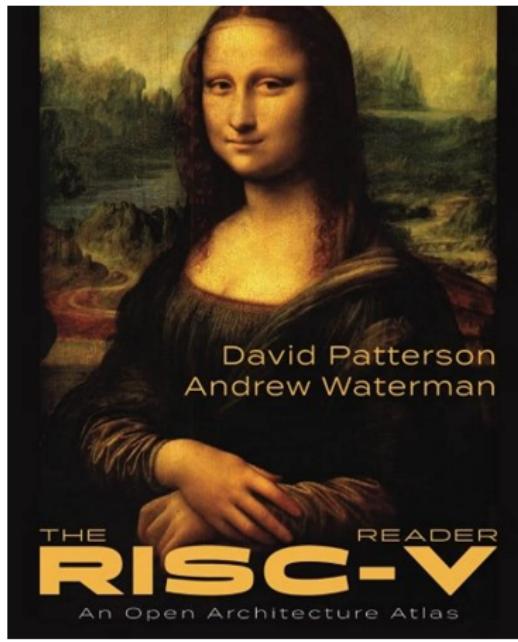


2019 年国际芯片大会



“RISC-V 很可能发展成为世界主流 CPU 之一，在 CPU 领域形成 Intel、ARM、RISC-V 三分天下的格局。” — 中国工程院院士 倪光南

大道至简





新长征路上的摇滚

“万里长征第一步 ...”

RISC-V Assembly Language Programming

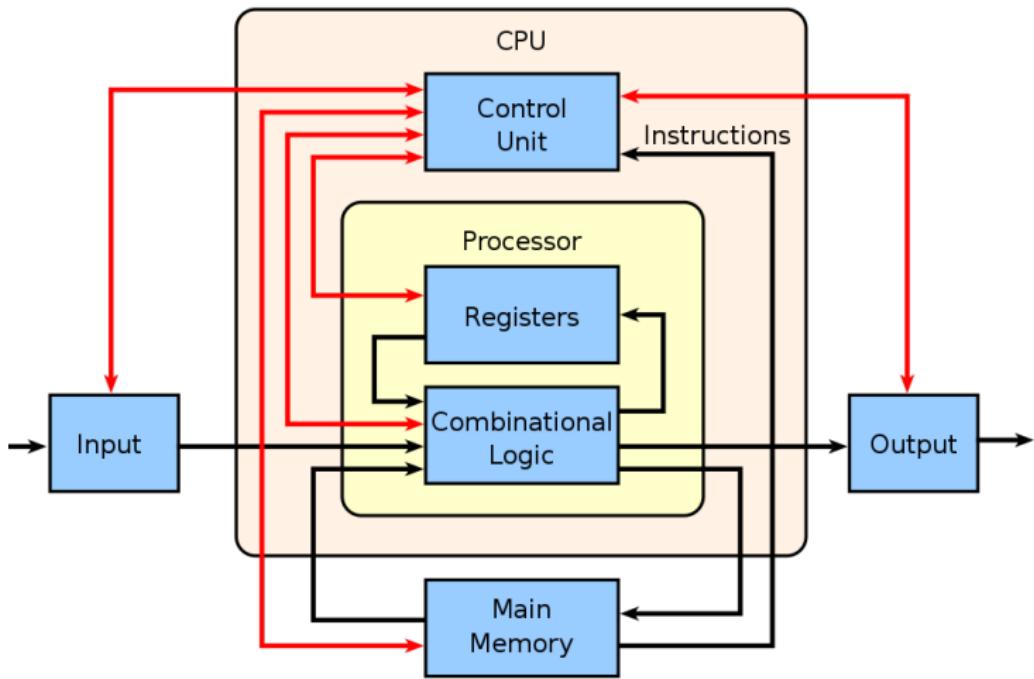
Using the ESP32-C3 and QEMU



Warren Gay



CONCEPT & CODE



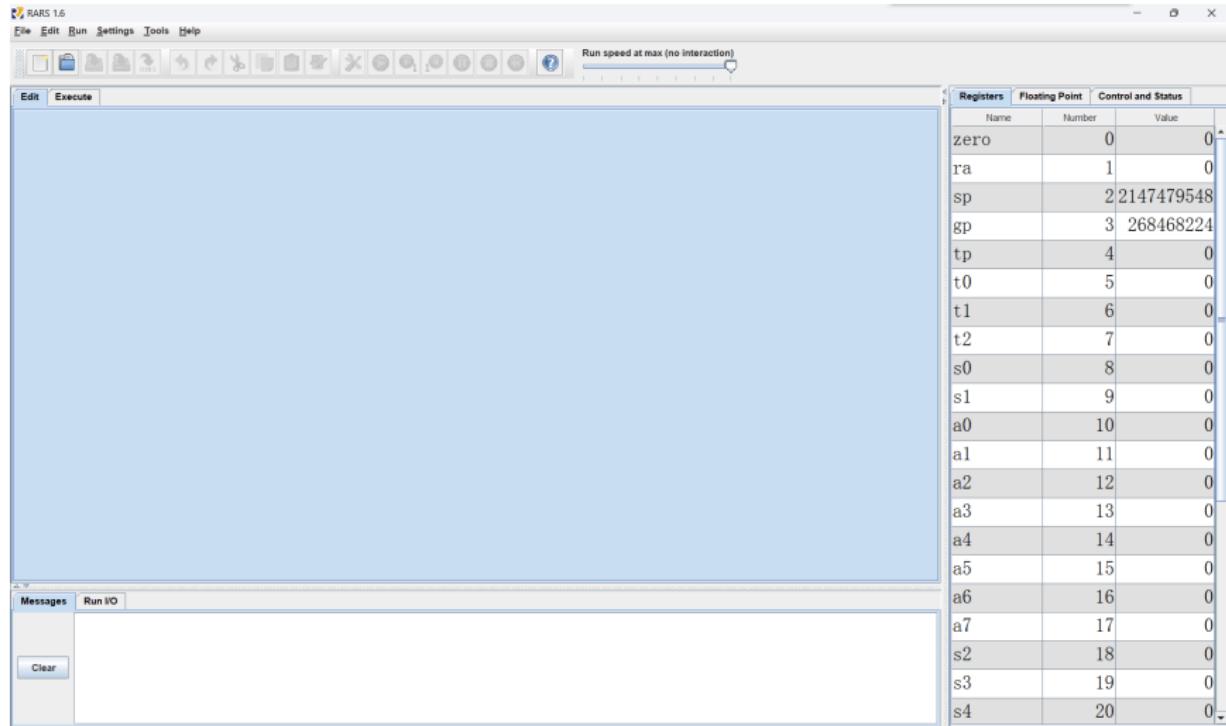
Registers (zero, pc)

Memory (Data, Instructions)

I/O



RARS: RISC-V Assembler and Runtime Simulator



<https://github.com/TheThirdOne/rars>

add.asm

add.asm

addi.asm

add.asm

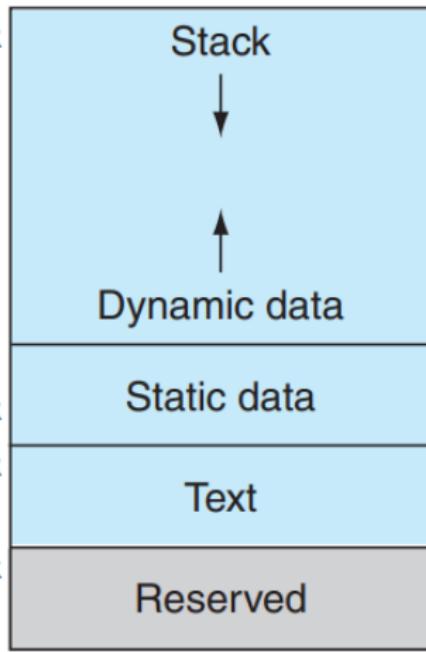
addi.asm

add-sub.asm

ecall.asm

data.asm

\$sp → 7fff fffc_{hex}



\$gp → 1000 8000_{hex}

1000 0000_{hex}

pc → 0040 0000_{hex}

0

array.asm

branch-max.asm

proc-max.asm



proc-max.asm



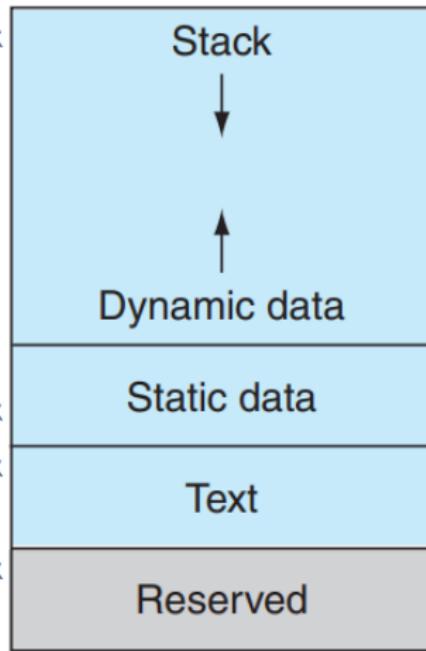
控制流 (control-flow) + 数据流 (data-flow)

$\$sp \rightarrow 7fff\ fffc_{hex}$

$\$gp \rightarrow 1000\ 8000_{hex}$

$1000\ 0000_{hex}$

$pc \rightarrow 0040\ 0000_{hex}$



proc-max.asm

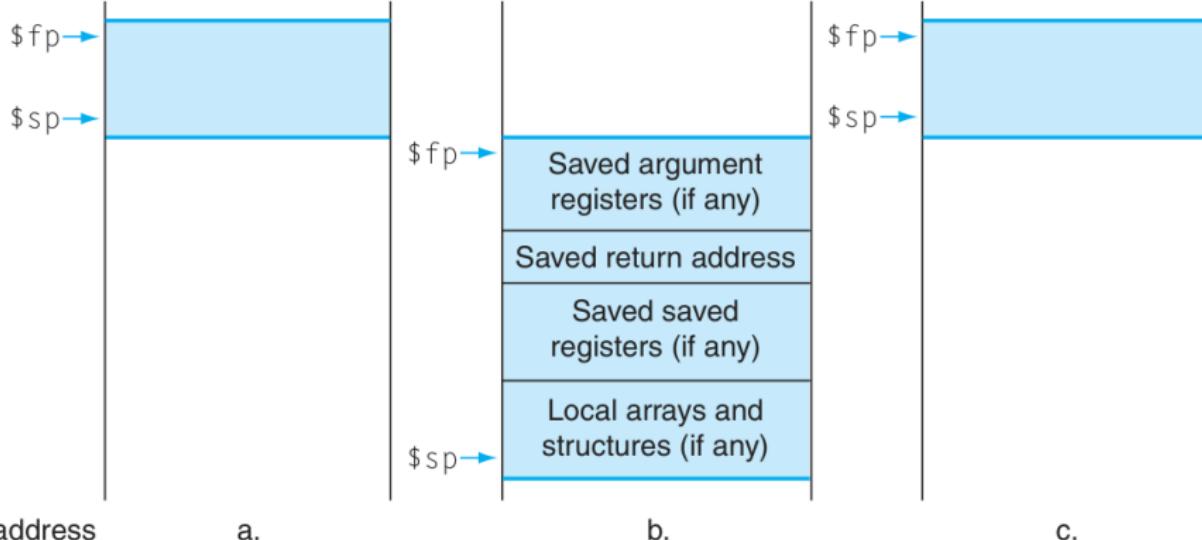


proc-fact.asm

RISC-V 调用约定 (Calling Convention)

Registers	Symbolic names	Description	Saver
x0	zero	Hardwired zero	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5-x7	t0-t2	Temporary registers	Caller
x8-x9	s0-s1	Saved registers	Callee
x10-x11	a0-a1	Function arguments and return values	Caller
x12-x17	a2-a7	Function arguments	Caller
x18-x27	s2-s11	Saved registers	Callee
x28-x31	t3-t6	Temporary registers	Caller

High address



Low address

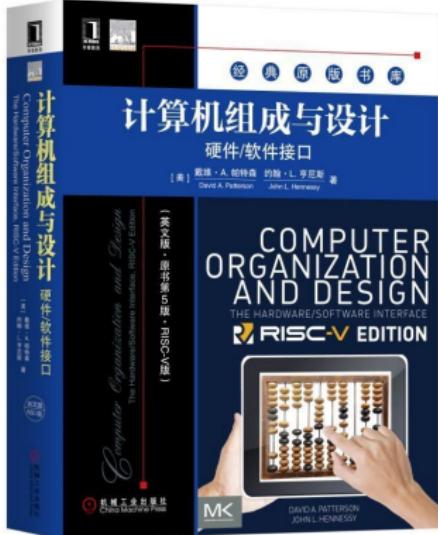
a.

b.

c.

bubblesort @ CompilerExplorer

tree-node @ CompilerExplorer

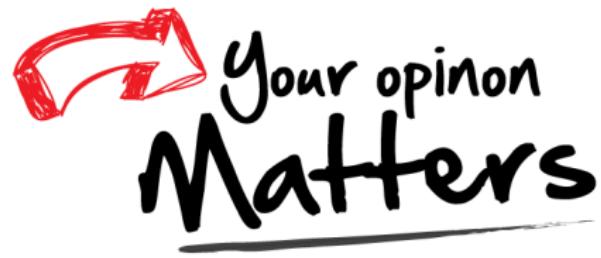


```
book-cover.s ●
Users > edson > Desktop > book-cover.s
1   .file  "book-cover.c"
2   .option nopic
3   .attribute arch, "rv32i2p0"
4   .attribute unaligned_access, 0
5   .attribute stack_align, 16
6   .text
7   .align 2
8   compute_the_answer_to_the_ultimate_question_of_life_the_universe_and_everything:
9     li a@,42
10    ret
11    .align 2
12    .globl do_something_1000_times
13    .type do_something_1000_times, @function
14 do_something_1000_times:
15    addi sp,sp,-16
16    sw $0,8(sp)
17    sw ra,12(sp)
18    li $0,1000
19 .L6:
20    addi $0,$0,-1
21    call do_something
22    bne $0,zero,.L6
23    lw ra,12(sp)
24    lw $0,8(sp)
25    addi sp,sp,16
26    jr ra
27    .section .rodata.str1.4,"aMS",@progbits,1
28    .align 2
29 .LC0:
30    .ascii "There are 10 types of people in this world "
31    .asciz "those who understand binary and those who don't"
32    .align 2
33 .LC1:
34    .string "Assembly language you must learn!"
35    .align 2
36 .LC2:
37    .ascii "The Unicamp CS course was created in 1969 - "
38    .asciz "The first one in Brazil!"
39 .LC3:
40    .byte 78, 105, 99, 101, 33, 32, 89, 111, 117, 32, 107
41    .byte 110, 111, 119, 32, 65, 83, 67, 73, 73, 33, 0
```

An Introduction to Assembly Programming with RISC-V

An Introduction to Assembly Programming with RISC-V

Thank You!



Office 926

hfwei@nju.edu.cn