



RISC-V и тестовые генераторы

МФТИ
Весна 2024

Кто?

- Тарасов Юлий [@botan razdolb](#)



- Белов Владислав [@sick hoof](#)



Где?

Аудитория – Цифра 3286

Слайды – Github [riscv-technologies-lab/testgen-lectures](https://github.com/riscv-technologies-lab/testgen-lectures)



Когда?

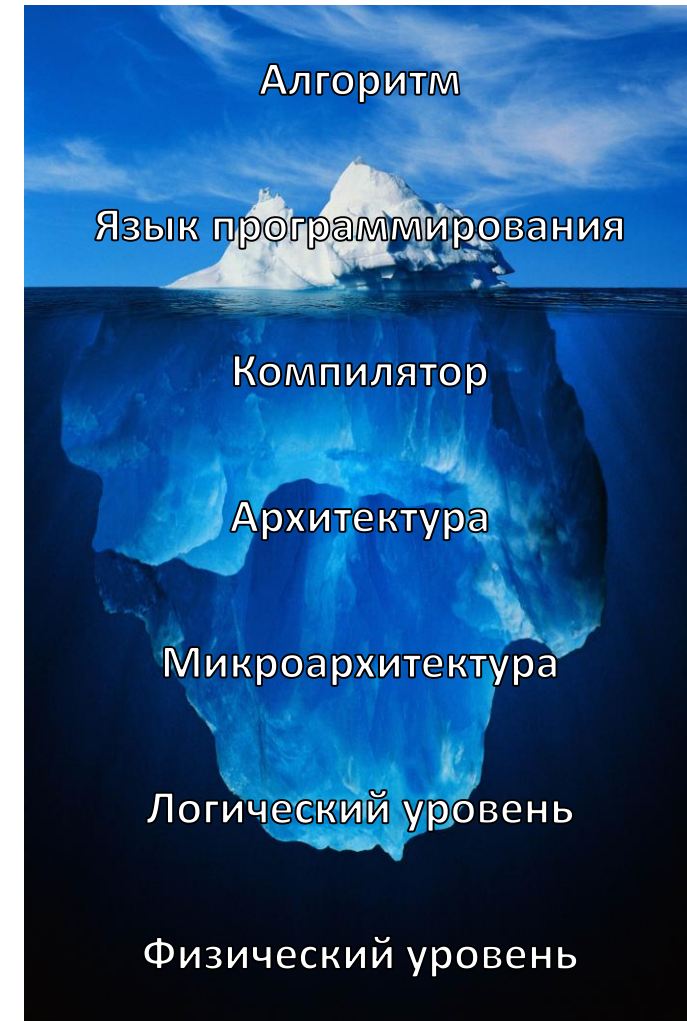
Понедельник 13:55-15:20

Что?

- Архитектура и экосистема RISC-V
- Как сделать свой RISC-V
- Как понять что ваш процессор работает (или нет)
- Настоящий тестовый генератор
- Как использовать генераторы
- Почему для успеха нужна модель
- Примеры отлова багов
- Веселое соревнование проектов

Уровни абстракции

- Абстракции помогают не сойти с ума
- SW – компилятор и выше
- HW – микроархитектура и ниже
- Архитектура – интерфейс между SW и HW



Архитектура определяет ...

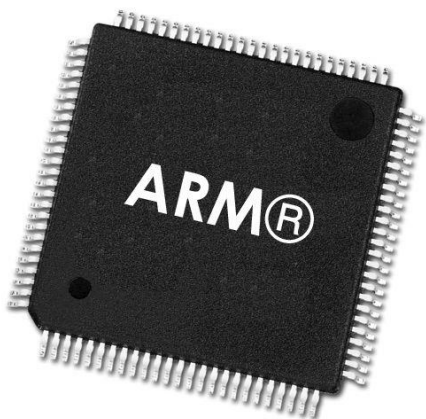
- Набор инструкций
- Регистры
- Типы данных
- Работу с памятью
- Ввод/вывод
- Создание прерываний и исключительных состояний и их обработка

Какие архитектуры уже есть



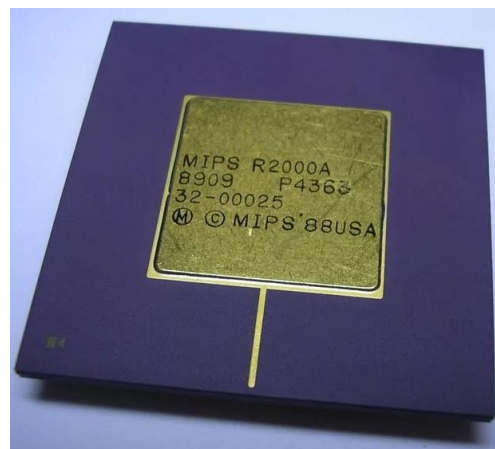
x86
x86-64

Ноутбуки, ПК, сервера



ARM

Мобильные устройства,
сервера



MIPS

PSP, сетевые устройства



Эльбрус

ДАННЫЕ УДАЛЕНЫ

Зачем еще одна?

КАК ПОЯВЛЯЮТСЯ СТАНДАРТЫ:



RISC-V



- Открытая
 - Спецификация в открытом доступе, royalty-free
- Свободная
 - Развитие в рамках комитета
- Модульная и расширяемая
 - Минимальный базовый набор
 - Дополнительная функциональность включается через расширения
 - Возможно включать расширения в различных комбинациях

RISC-V Foundation

- Основан в 2015
индустриальными лидерами
и стартапами
- 3900+ членов из 70+ стран
- Продвигает исследования и
инновации

AKEANA

Alibaba Cloud

ANDES
TECHNOLOGY

OS
北京开源芯片研究院
BEIJING INSTITUTE OF OPEN SOURCE CHIP

成为资本 CHENGWEI
CAPITAL

Google

HUAWEI

ICT
FOUNDING

Imagination

ISCAS

intel.

NVIDIA

Phytium 飞腾

Qualcomm

RIOS

Rivos

ZTE

SEAGATE

siFive

希姆计算
STREAM COMPUTING

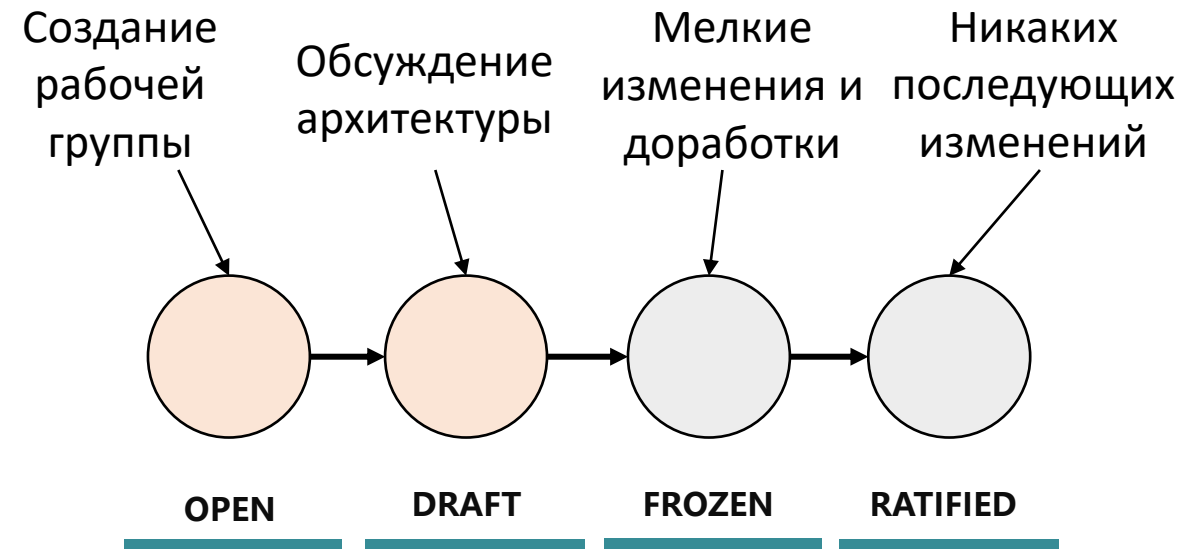
SYNOPSYS

Syntacore
Custom cores and tools

Tencent
腾讯

VENTANA
MICRO

Процесс ратификации спецификации RISC-V



Альянс RISC-V

Задачи:

- создание открытого сообщества разработчиков
- участие в фундаментальных исследованиях
- развитие российской экосистемы продуктов

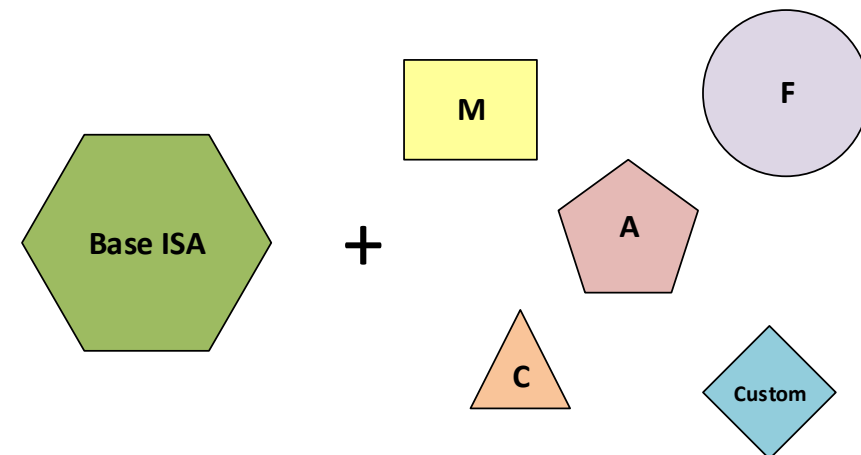
Профильные комитеты:

- Технологический
- Индустриальный
- Юридический
- Академический

Модульность и расширяемость RISC-V

- Базовый набор и стандартные расширения зафиксированы
- Добавление функциональности через расширения, не выпуск новых версий

M	Умножение и деление
C	Сжатые инструкции
F	Single-precision floats
D	Double-precision floats
E	Сокращенное количество регистров
A	Атомики
Z*	Другие стандартные расширения



Минимализм базовой ISA

Jumps & Calls	Loads & Stores	Arithmetics				Special
JAL	LB	ADD	ADDI	ADDW	ADDIW	FENCE
JALR	LH	SUB		SUBW		ECALL
BEQ	LW	OR	ORI			EBREAK
BNE	LBU	XOR	XORI			
BLT	LHU	AND	ANDI			
BGE	SB	SRL	SRLI	SRLW	SRLIW	
BLTU	SH	SLL	SLLI	SLLW	SLLIW	Upper immediate
BGEU	SW	SRA	SRAI	SRAW	SRAIW	LUI
	LWU	Data flow				AIUPC
	LD	SLT	SLTU	SLTI	SLTIU	
	SD					

Псевдооперации

```
for (i = 0; i < N; ++i)
    if (a[i] == x)
        return i;
```

```
        mv        a5,a0
        li        a0,0
        j         .loop
.latch:
        addiw     a0,a0,1
.loop:
        lw        a4,0(a5)
        addi      a5,a5,4
        bne      a4,a2,.latch
.exit:
        ret
```

Псевдоинструкция	Базовая инструкция	Смысл
nop	addi x0, x0, 0	Нет операции
li rd, immediate	<i>Различные последовательности</i>	Загрузка константы
mv rd, rs	addi rd, rs, 0	Копирование регистров
not rd, rs	xori rd, rs, -1	Инверсия числа
neg rd, rs	sub rd, x0, rs	Изменение знака числа
seqz rd, rs	sltiu rd, rs, 1	Установить 1, если == 0
snez rd, rs	sltu rd, x0, rs	Установить 1, если != 0
sltz rd, rs	slt rd, rs, x0	Установить 1, если < 0
sgtz rd, rs	slt rd, x0, rs	Установить 1, если > 0
beqz rs, offset	beq rs, x0, offset	Перейти, если == 0
bnez rs, offset	bne rs, x0, offset	Перейти, если != 0
blez rs, offset	bge x0, rs, offset	Перейти, если <= 0
bgez rs, offset	bge rs, x0, offset	Перейти, если >= 0
bltz rs, offset	blt rs, x0, offset	Перейти, если < 0
bgtz rs, offset	blt x0, rs, offset	Перейти, если > 0
bgt rs1, rs2, offset	blt rs2, rs1, offset	Перейти, если >
ble rs1, rs2, offset	bge rs2, rs1, offset	Перейти, если <=
bgtu rs1, rs2, offset	bltu rs2, rs1, offset	Перейти, если >, беззнаковое
bleu rs1, rs2, offset	bgeu rs2, rs1, offset	Перейти, если <=, беззнаковое
j offset	jal x0, offset	Переход по метке
jal offset	jal x1, offset	Переход с сохранением адреса возврата
jr rs	jalr x0, 0(rs)	Переход по значению из регистра
jalr rs	jalr x1, 0(rs)	Переход с сохранением адреса возврата
ret	jalr x0, x1, 0	Возврат из подпрограммы

Формат кодировки

31	30	25	24	21	20	19	15	14	12	11	8	7	6	0				
funct7				rs2			rs1		funct3		rd			opcode		R-type		
imm[11:0]						rs1		funct3		rd			opcode		I-type			
imm[11:5]				rs2			rs1		funct3		imm[4:0]			opcode		S-type		
imm[12]		imm[10:5]			rs2			rs1		funct3		imm[4:1]		imm[11]		opcode		B-type
imm[31:12]										rd			opcode			U-type		
imm[20]		imm[10:1]				imm[11]		imm[19:12]			rd			opcode		J-type		

ABI и calling convention

Name	Alias
x0	zero
x1	ra
x2	sp
x3	gp
x4	tp
x5-x7	t0-t2
x8-x9	s0-s1
x10-x15, x16, x17	a0-a5, a6, a7
x18-x27	s2-s11
x28-x31	t3-t6

FP Name	FP Alias
f0-f7	ft0-ft7
f8-f9	fs0-fs1
f10-f17	fa0-fa7
f18-f27	fs2-fs11
f28-f31	ft8-ft11

Красных регистров нет в RV32E

Обсуждение

Оказавшись компилятором, чтобы вы сказали?

- Для RV32I?
- Для RV32IF?
- Для RV32ID?

```
int fto_int(float f) {  
    return f;  
}
```

Гладко было на бумаге...

Не компилируется, потому что нет fp регистров ... но это можно заставить скомпилироваться: godbolt.org

Доступные RISC-V ABI: **ilp32 ilp32d ilp32e ilp32f lp64 lp64d lp64f**

LP64 – long и pointer = 64, int = 32

LP64d – то же и поддерживаны double

ILP32, ILP32d – integer, long и pointer = 32

А с каким ABI собрана libc с которой вы линкуетесь? :-)

To be continued ...

На следующем занятии прикоснемся к экосистеме RISC-V:

- Научимся пользоваться кросс-компилятором (и узнаем почему он кросс)
- Научимся запускать RISC-V программы без RISC-V
- Запустим программу на настоящем RISC-V
- Научимся отлаживаться удаленно

Канал курса в telegram

