

Введение в архитектуру RISC-V Симулятор RARS

Дмитрий Смехов

Никита Поляков - Доклад на ChipExpo 2021:

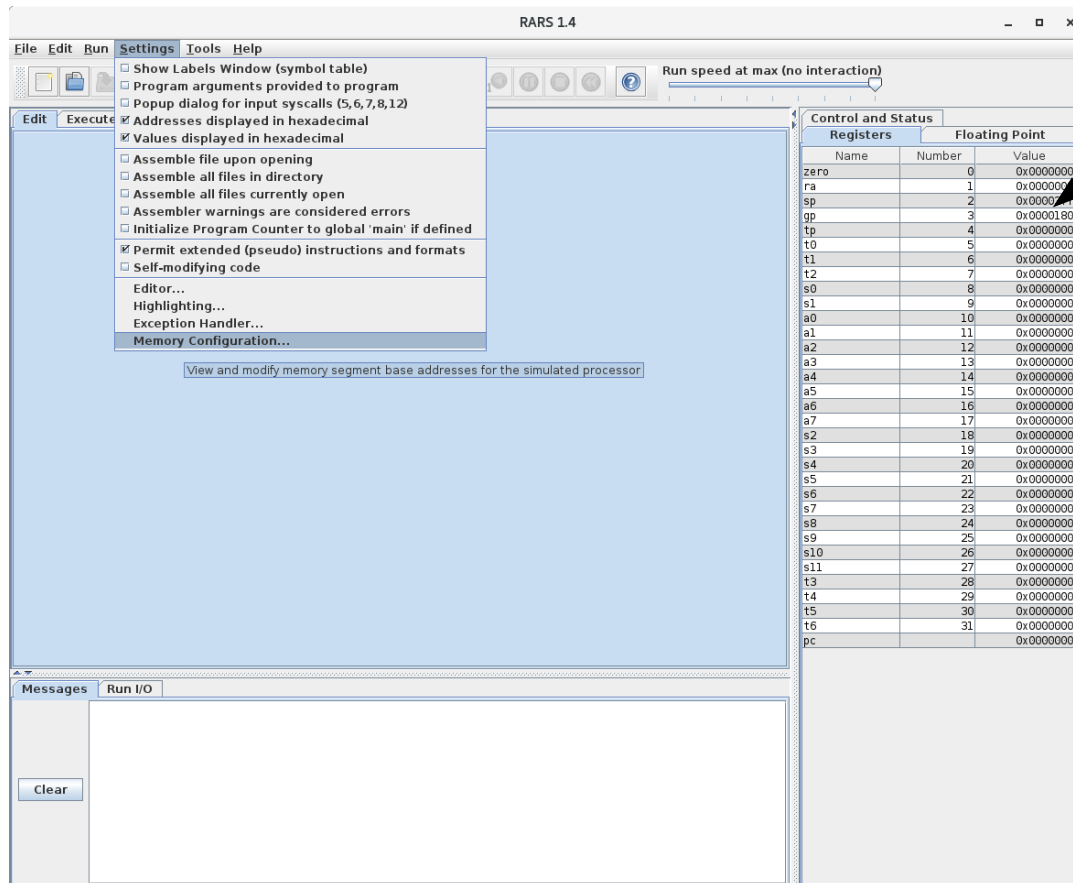
«Введение в архитектуру RISC-V. Язык ассемблера».

<https://youtu.be/3jBVzEKVgA4>

Запуск RARS

Перейти в каталог
lab_day3_01_rars

Выполнить
./run_rars.sh &



Область регистров

- 32 регистра для основного набора команд
- каждый регистр имеет размер 32 бита = слово (word)
- x0 всегда равен 0
- 32 регистра для вещественных операций в расширении “F”
- в версиях RV64 регистры имеют размер 64 бита (double word)

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	t0	Temporary/alternate link register	Caller
x6–7	t1–2	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

Модель памяти

Memory Configuration

Configuration

☐ Default

☐ Compact, Data at Address 0

☒ Compact, Text at Address 0

0x00007fff	memory map limit address
0x00007fff	kernel space high address
0x00007f00	MMIO base address
0x00004000	kernel space base address
0x00003fff	user space high address
0x00003fff	data segment limit address
0x00003ffc	stack pointer (sp)
0x00003ffc	stack base address
0x00003000	stack limit address
0x00003000	heap base address
0x00002000	.data base address
0x00001800	global pointer (gp)
0x00001000	data segment base address
0x00001000	.extern base address
0x00000ffc	text limit address
0x00000000	.text base address

Apply and Close Apply Cancel Reset

Три модели памяти:

- Compact, Text at Address 0 — 32 кБ
- Compact, Data at Address 0 — 32 кБ
- Default — 4 ГБ

Первая программа

Программа src/task1.S

Команды:

li rd, imm

addi rd, rs1, imm

beq rs1, rs2, label

Обозначения:

rs1 - регистр источника 1

rs2 - регистр источника 2

rd — регистр назначения

label — метка в программе

imm — константа размером 12 бит

```
task1.S
1  # RISC-V simple program
2  #
3
4          .text
5
6          li    a0, 0x55
7          addi  a0, zero, 0x55
8
9          li    t6, 2
10         li    t5, 0x10
11
12
13 start:   addi  a0, a0, 1
14         beq   zero, zero, start
15
```

Edit
Execute

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00000000	0x05500513	addi x10,x0,0x00000055	6: li a0, 0x55
<input type="checkbox"/>	0x00000004	0x05500513	addi x10,x0,0x00000055	7: addi a0, zero, 0x55
<input type="checkbox"/>	0x00000008	0x00200f93	addi x31,x0,0x00000002	9: li t6, 2
<input type="checkbox"/>	0x0000000c	0x01000f13	addi x30,x0,0x00000010	10: li t5, 0x10
<input type="checkbox"/>	0x00000010	0x00150513	addi x10,x10,0x0000...	13: start: addi a0, a0, 1
<input type="checkbox"/>	0x00000014	0xfe000ee3	beq x0,x0,0xffffffffe	14: beq zero, zero, start

Control and Status

Registers

Floating Point

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x00003ffc
gp	3	0x00001800
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000055
a1	11	0x00000000
a2	12	0x00000000

F3 — компиляция кода

F7 — один шаг выполнения программы

Code — машинный код. Обратите внимание — первые две команды имеют один и тот же машинный код

Выполните несколько шагов программы.

Измените программу для увеличения регистра t6 на 2

Вторая программа

Программа src/task2.S

Команды:

sw rd, offset

add rd, rs1, rs2

bne rs1, rs2, label

.text — сегмент программы

.data — сегмент данных

.word — резервирование слова в памяти

task2.S

```
1  # RISC-V simple program
2  #
3
4      .text
5      la    a3, var0
6      li    a0, 0x55
7      li    a1, 0x11
8      add   a2, a0, a1
9      sw    a2, 0(a3)
10
11     li    t6, 4
12
13 start:  addi a2, a2, 1
14         addi t6, t6, -1
15         bne zero, t6, start
16
17         sw a2, 4(a3)
18 stop:   beq zero, zero, stop
19
20     .data
21 var0:   .word 0
22         .word 0xFFFFFFFF
23
```


Выполнение программы 2

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00000000	0x00002697	auipc x13,0x00000002	5: la a3, var0
<input type="checkbox"/>	0x00000004	0x00068693	addi x13,x13,0x0000...	
<input type="checkbox"/>	0x00000008	0x05500513	addi x10,x0,0x00000055	6: li a0, 0x55
<input type="checkbox"/>	0x0000000c	0x01100593	addi x11,x0,0x00000011	7: li a1, 0x11
<input type="checkbox"/>	0x00000010	0x00b50633	add x12,x10,x11	8: add a2, a0, a1
<input type="checkbox"/>	0x00000014	0x00c6a023	sw x12,0x00000000(x13)	9: sw a2, 0(a3)
<input type="checkbox"/>	0x00000018	0x00400f93	addi x31,x0,0x00000004	11: li t6, 4
<input type="checkbox"/>	0x0000001c	0x00160613	addi x12,x12,0x0000...	13: start: addi a2, a2, 1
<input type="checkbox"/>	0x00000020	0xffff8f93	addi x31,x31,0xffff...	14: addi t6, t6, -1
<input type="checkbox"/>	0x00000024	0xfff01ce3	bne x0,x31,0xffffffffc	15: bne zero, t6, start
<input type="checkbox"/>	0x00000028	0x00c6a223	sw x12,0x00000004(x13)	17: sw a2, 4(a3)
<input type="checkbox"/>	0x0000002c	0x00000063	beq x0,x0,0x00000000	18: stop: beq zero, zero, stop

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x00002000	0x00000000	0xffffffff	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00002020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00002040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Registers

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x00003ffc
gp	3	0x00001800
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00002000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000

Первая команда — загрузка адреса, две команды в машинном коде
Регистр A3 содержит адрес метки **var0**, это начало секции **.data**, 0x2000

Адрес 0x2004 — содержит значение 0xFFFFFFFF, то что задано в тексте программы