Extensibility

Extensibility

Translators are easy to add

```
pub trait Translator {
  fn variable_info(&self, variable: &VariableMeta)
   -> Result<VariableInfo>;
  fn translate(&self, variable, value)
   -> Result<TranslationResult>;
  fn translates(&self, variable: &VariableMeta)
   -> Result<TranslationPreference>;
}
```

Extensibility

Translators are easy to add

```
pub trait Translator {
  fn variable_info(&self, variable: &VariableMeta)
   -> Result<VariableInfo>;
  fn translate(&self, variable, value)
   -> Result<TranslationResult>;
  fn translates(&self, variable: &VariableMeta)
   -> Result<TranslationPreference>;
}
```

Extensibility

Translators are easy to add

```
pub trait Translator {
  fn variable_info(&self, variable: &VariableMeta)
   -> Result<VariableInfo>;
  fn translate(&self, variable, value)
   -> Result<TranslationResult>;
  fn translates(&self, variable: &VariableMeta)
   -> Result<TranslationPreference>;
}
```

Extensibility

Translators are easy to add

```
pub trait Translator {
  fn variable_info(&self, variable: &VariableMeta)
   -> Result<VariableInfo>;
  fn translate(&self, variable, value)
   -> Result<TranslationResult>;
  fn translates(&self, variable: &VariableMeta)
   -> Result<TranslationPreference>;
}
```

Extensibility

Supports multiple waveform sources

VCD, FST, GHW

Or completely different formats

- TLM transaction streams

Interactively via TCP

Remote Control

Remote Control Protocol via TCP, (Web)Sockets, …

Full control over the waveform viewer

- Add or remove signals
- Zoom or pan around
- Mark parts, draw inside Waveform

Runs everywhere

Linux, Mac and Windows of course.

Web Assembly!

Web Assembly!

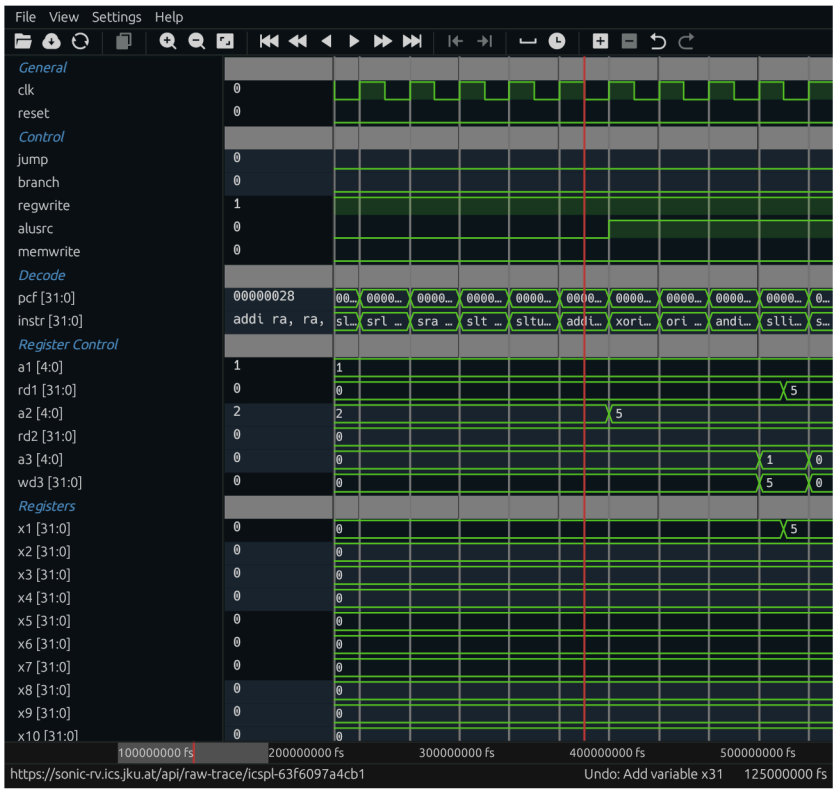
- No installation
- Inspect waves in continuous integration
  - VUnit exaple: https://oscargus.github.io/fulladder/
- **Embeddable**

Embeddable

Sonic RV | Examples | Editor | CPU | Waveform | About

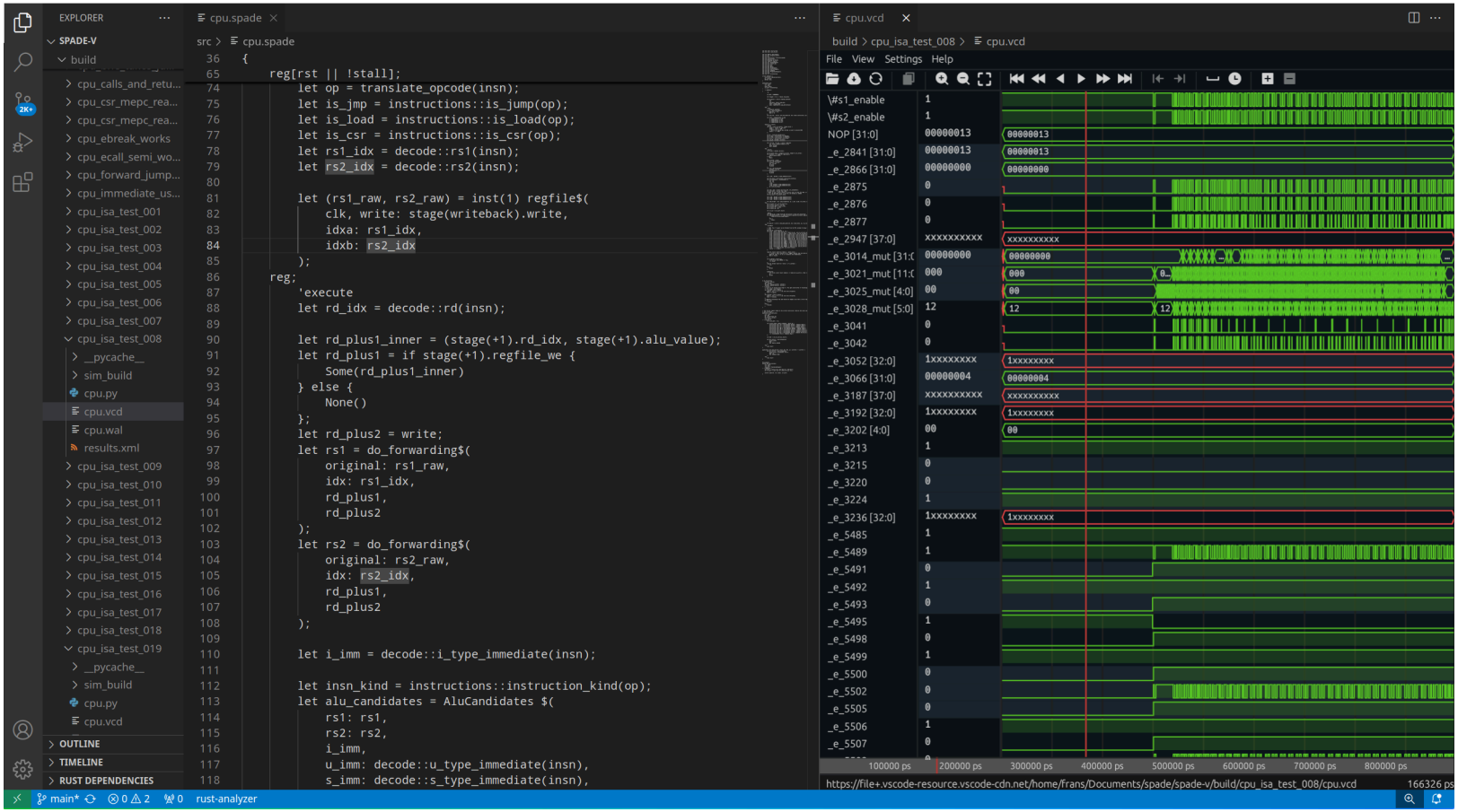
Steps

PC	fetch	decode	execute	memory	writeback
0	add x0, x1, x2				
4	sub x0, x1, x2	add x0, x1, x2			
8	xor x0, x1, x2	sub x0, x1, x2	add x0, x1, x2		
C	or x0, x1, x2	xor x0, x1, x2	sub x0, x1, x2	add x0, x1, x2	
10	and x0, x1, x2	or x0, x1, x2	xor x0, x1, x2	sub x0, x1, x2	add x0, x1, x2
14	sll x0, x1, x2	and x0, x1, x2	or x0, x1, x2	xor x0, x1, x2	sub x0, x1, x2
18	srl x0, x1, x2	sll x0, x1, x2	and x0, x1, x2	or x0, x1, x2	xor x0, x1, x2
1C	sra x0, x1, x2	srl x0, x1, x2	sll x0, x1, x2	and x0, x1, x2	or x0, x1, x2
20	slt x0, x1, x2	sra x0, x1, x2	srl x0, x1, x2	sll x0, x1, x2	and x0, x1, x2
24	sltu x0, x1, x2	slt x0, x1, x2	sra x0, x1, x2	srl x0, x1, x2	sll x0, x1, x2
28	addi x1, x1, 5	sltu x0, x1, x2	slt x0, x1, x2	sra x0, x1, x2	srl x0, x1, x2
2C	xori x0, x1, 5	addi x1, x1, 5	sltu x0, x1, x2	slt x0, x1, x2	sra x0, x1, x2
30	ori x0, x1, 5	xori x0, x1, 5	addi x1, x1, 5	sltu x0, x1, x2	slt x0, x1, x2
34	andi x0, x1, 5	ori x0, x1, 5	xori x0, x1, 5	addi x1, x1, 5	sltu x0, x1, x2
38	slli x0, x1, 0x05	andi x0, x1, 5	ori x0, x1, 5	xori x0, x1, 5	addi x1, x1, 5



Institute for Complex Systems | Johannes Kepler University

Embeddable


EXPLORER
SPADE-V
build
cpu_calls_and_retu...
cpu_csr_mepc_rea...
cpu_csr_mepc_rea...
cpu_ebreak_works
cpu_ecall_semi_wo...
cpu_forward_jump...
cpu_immediate_us...
cpu_isa_test_001
cpu_isa_test_002
cpu_isa_test_003
cpu_isa_test_004
cpu_isa_test_005
cpu_isa_test_006
cpu_isa_test_007
cpu_isa_test_008
__pycache__
sim_build
cpu.py
cpu.vcd
cpu.wal
results.xml
cpu_isa_test_009
cpu_isa_test_010
cpu_isa_test_011
cpu_isa_test_012
cpu_isa_test_013
cpu_isa_test_014
cpu_isa_test_015
cpu_isa_test_016
cpu_isa_test_017
cpu_isa_test_018
cpu_isa_test_019
__pycache__
sim_build
cpu.py
cpu.vcd
OUTLINE
TIMELINE
RUST DEPENDENCIES
cpu.spade
src > cpu.spade
36 {
65 reg[rst || !stall];
74 let op = translate_opcode(insn);
75 let is_jmp = instructions::is_jump(op);
76 let is_load = instructions::is_load(op);
77 let is_csr = instructions::is_csr(op);
78 let rs1_idx = decode::rs1(insn);
79 let rs2_idx = decode::rs2(insn);
80
81 let (rs1_raw, rs2_raw) = inst(1) regfile$(
82 clk, write: stage(writeback).write,
83 idxa: rs1_idx,
84 idxb: rs2_idx
85);
86 reg;
87 'execute
88 let rd_idx = decode::rd(insn);
89
90 let rd_plus1_inner = (stage(+1).rd_idx, stage(+1).alu_value);
91 let rd_plus1 = if stage(+1).regfile_we {
92 Some(rd_plus1_inner)
93 } else {
94 None()
95 };
96 let rd_plus2 = write;
97 let rs1 = do_forwarding$(
98 original: rs1_raw,
99 idx: rs1_idx,
100 rd_plus1,
101 rd_plus2
102);
103 let rs2 = do_forwarding$(
104 original: rs2_raw,
105 idx: rs2_idx,
106 rd_plus1,
107 rd_plus2
108);
109
110 let i_imm = decode::i_type_immediate(insn);
111
112 let insn_kind = instructions::instruction_kind(op);
113 let alu_candidates = AluCandidates $(
114 rs1: rs1,
115 rs2: rs2,
116 i_imm,
117 u_imm: decode::u_type_immediate(insn),
118 s_imm: decode::s_type_immediate(insn),
cpu.vcd
build > cpu_isa_test_008 > cpu.vcd
File View Settings Help
\#s1_enable
\#s2_enable
NOP [31:0]
_e_2841 [31:0]
_e_2866 [31:0]
_e_2875
_e_2876
_e_2877
_e_2947 [37:0]
_e_3014_mut [31:0]
_e_3021_mut [11:0]
_e_3025_mut [4:0]
_e_3028_mut [5:0]
_e_3041
_e_3042
_e_3052 [32:0]
_e_3066 [31:0]
_e_3187 [37:0]
_e_3192 [32:0]
_e_3202 [4:0]
_e_3213
_e_3215
_e_3220
_e_3224
_e_3236 [32:0]
_e_5485
_e_5489
_e_5491
_e_5492
_e_5493
_e_5495
_e_5498
_e_5499
_e_5500
_e_5502
_e_5505
_e_5506
_e_5507
100000 ps
200000 ps
300000 ps
400000 ps
500000 ps
600000 ps
700000 ps
800000 ps
https://file+.vscode-resource.vscode-cdn.net/home/frans/Documents/spade/spade-v/build/cpu_isa_test_008/cpu.vcd
166326 ps
main*
0 2
0
rust-analyzer

Integrated Analysis

Analysis features inside Surfer with WAL

- Via TCP
- Long term goal: WAL directly inside Surfer

Performance Analysis

Inject new logic inside Surfer

- Create little helpers
- ready && valid
- req |-> ##[1:2] ack
- Create abstractions for buses

Some technical details

Written in Rust

- Nearly Free Web Assembly support

Some technical details

Written in Rust

- (Perhaps too) Easy dependency management

```toml
[dependencies]
asm_riscv = "0.2.0"
base64 = "0.22"
bytes = "1.5.0"
camino = { version = "1.1.6", features = ["serde1"] }
chrono = "0.4.31"
clap = { version = "4.4.10", features = ['derive'] }
color-eyre = "0.6.2"
config = { version = "0.14", default_features = false, features = ["toml"] }
derivative = "2.2.0"
derive_more = "0.99.17"
directories = "5.0"
eframe = "0.25.0"
egui-remixicon = "0.2"
egui_extras = "0.25.0"
egui_plot = { version = "0.25", optional = true }
enum-iterator = "2.0"
f128 = { path = "f128", optional = true }
fern = { version = "0.6.2", features = ["colored"] }
futures = { version = "0.3.29", features = ["executor"] }
futures-core = "0.3.29"
futures-util = "0.3.29"
fuzzy-matcher = "0.3.7"
fzcmd = { path = "fzcmd" }
half = "2.3.1"
human-sort = "0.2.2"
itertools = "0.12.0"
lazy_static = "1.4.0"
log = "0.4"
num = { version = "0.4", features = ["serde"] }
progress-streams = "1.1.0"
pure-rust-locales = "0.8.1"
rayon = "1.8.0"
regex = "1.10.2"
reqwest = { version = "0.11.22", features = ["stream"] }
rfd = { version = "0.14.0", default-features = false, features = ["tokio", "xdg-portal"] }
ron = { version = "0.8.1", features = ["integer128"] }
serde = { version = "1.0.193", features = ["derive"] }
serde_json = "1.0.113"
serde_stacker = "0.1"
softposit = "0.4.0"
spade = { path = "spade/spade-compiler", optional = true }
spade-common = { path = "spade/spade-common", optional = true }
spade-hir-lowering = { path = "spade/spade-hir-lowering", optional = true }
spade-mir = { path = "spade/spade-mir", optional = true }
spade-types = { path = "spade/spade-types", optional = true }
sys-locale = "0.3.1"
tokio = { version = "1.35.1", features = ["rt", "time", "macros", "sync"] }
toml = "0.8.8"
vcd-translate = { path = "spade/vcd-translate", optional = true }
web-sys = { version = "0.3.66", features = ["Location", "UrlSearchParams"] }
wellen = { path = "wellen" }
```

Waveform Backends

We don’t do the wave parsing ourselves

FastWaveBackend - Yehowshua Immanuel

Waveform Backends

We needed FST support. Kevin Laeufer asked me if I wanted a Rust library for it

Waveform Backends

We needed FST support. Kevin Laeufer asked me if I wanted a Rust library for it

Enter **Wellen**

- Multi-threaded loading
  - 2 minutes reduced to 6 seconds
- On-demand decompression
  - Large VCDs use much less memory

Surver

- Host waveforms on your simulator server
- Look at them on your dev-machine
  - Only signals you view are donwloaded

Thanks!

Contributors

- **Yehowshua Immanuel**
- **Kevin Laeufer**
- **Andreas Wallner**
- **Matt Taylor**
- **Francesco Urbani**
- **Tom Verbeure**
- **Hugo Lundin**
- **Verneri Hirvonen**
- **Paul Blume**
- **Felix Roithmayr**
- **Christian Dattinger**
- **Alejandro Tafalla**
- **Kacper Uminski**
- **James Connolly**
- **Lars Kadel**
- **Daniel Große**

Everyone who has provided feedback

Conclusions and A call for action

Surfer is a **snappy** and **extensible** waveform viewer that **runs everywhere**

Conclusions and A call for action

Surfer is a **snappy** and **extensible** waveform viewer that **runs everywhere**

What do **you** want from a waveform viewer?

Conclusions and A call for action

Surfer is a **snappy** and **extensible** waveform viewer that **runs everywhere**

What do **you** want from a waveform viewer?

Try it on your phone!

LINKÖPING UNIVERSITY