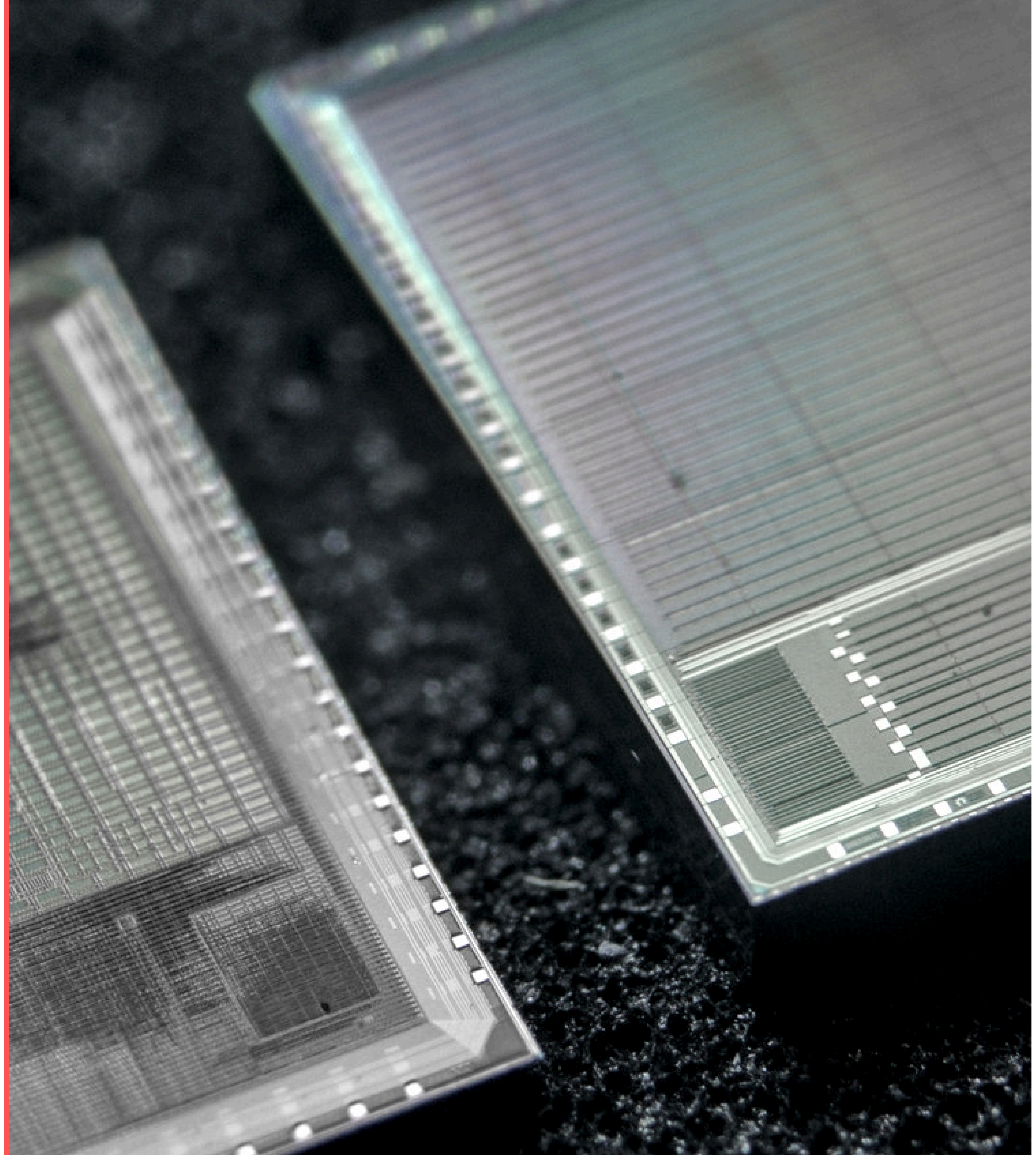
Collaborate, Customize, Innovate:

Open Source EDA unlocks Productivity

Frans Skarman and Stefan Wallentowitz
HM Munich University of Applied Sciences
HIPEAC 2026



Productivity as a Key Challenge

- Productivity gains are critical for Europe to catch up globally
- Chip designers must be able to
  - make best use of their precious time (skill shortage)
  - easily work with various tools in different flows
  - extend and augment flows to their needs
  - differentiate on their own core business

Open Source EDA as Key Ingredient for Productivity

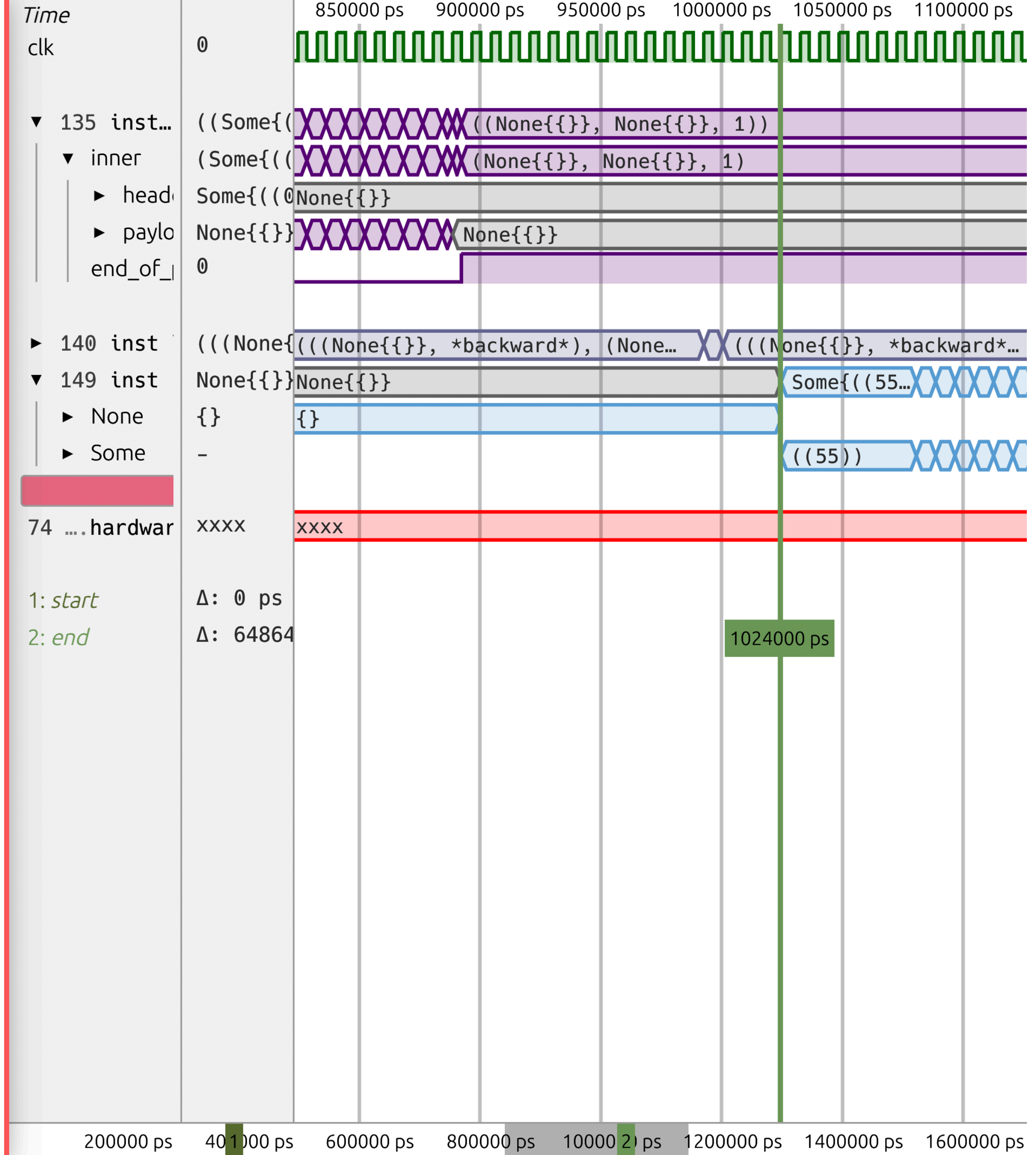
- Rich open source ecosystem
  - provides **point tools** for specific needs
  - allows to **modify tools** to match use cases
  - nurtures **novel** methodologies
  - incentivizes **curiosity** and **innovative** thinking
- Productivity, integration and verification tooling as focus areas
- In the following: Examples for productivity focused tools

Examples addressed in ODE4EC-PIV

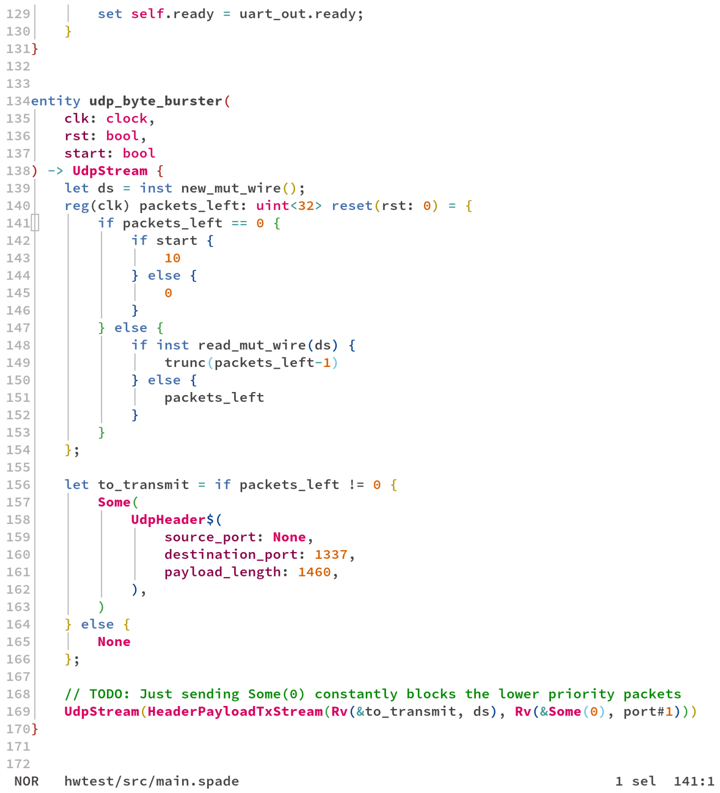
- Modular Frameworks for Design Entry and Flow Integration
- Industry-Standard Verification
- AI-Driven Novel Methodologies for Hardware Design

In the following: Examples for productivity focused tools that help to **innovate**, **customize** and **collaborate**

Surfer

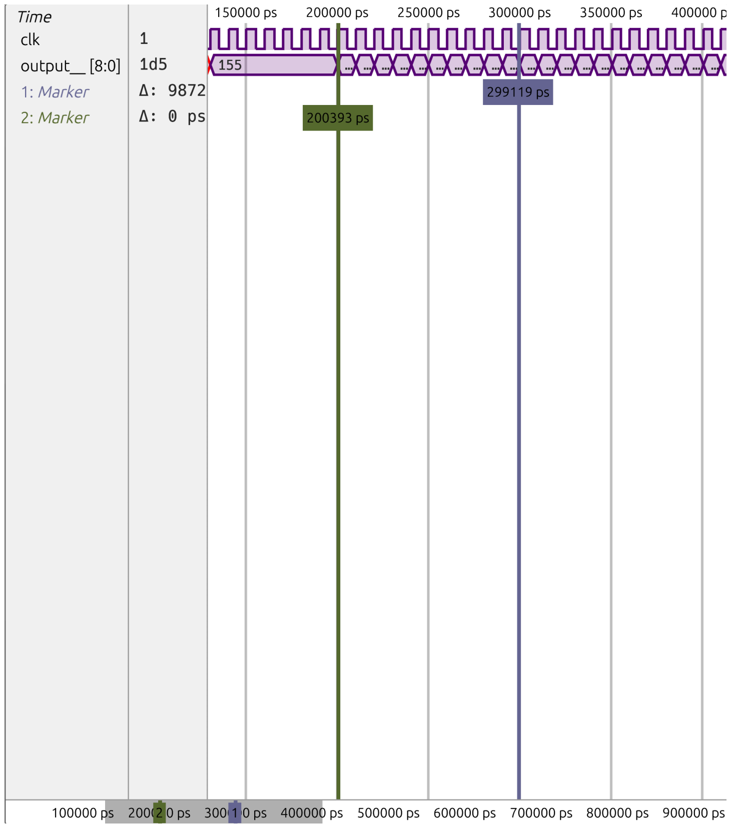

Time
850000 ps
900000 ps
950000 ps
1000000 ps
1050000 ps
1100000 ps
clk
0
▼ 135 inst…
((Some{(
((None{{}}, None{{}}, 1))
▼ inner
(Some{((
(None{{}}, None{{}}, 1)
▶ head
Some{((0
None{{}}
▶ paylo
None{{}}
None{{}}
end_of_
0
▶ 140 inst
(((None{
(((None{{}}, *backward*), (None…
(((None{{}}, *backward*…
▼ 149 inst
None{{}}
None{{}}
Some{((55…
▶ None
{}
{}
▶ Some
-
((55))
74 ….hardwar
xxxx
xxxx
1: start
Δ: 0 ps
2: end
Δ: 64864
1024000 ps
200000 ps
400000 ps
600000 ps
800000 ps
1000000 ps
1200000 ps
1400000 ps
1600000 ps

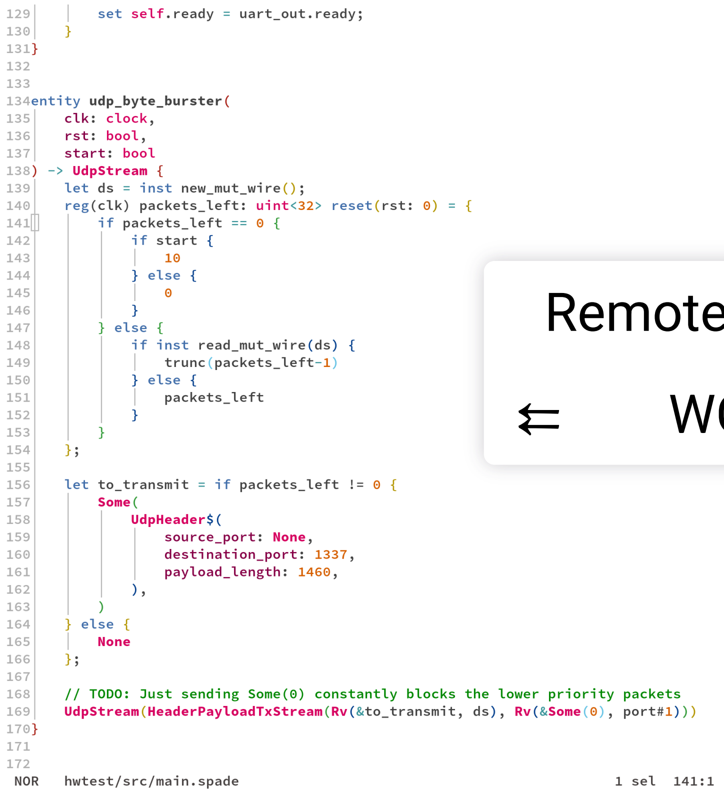
Innovate



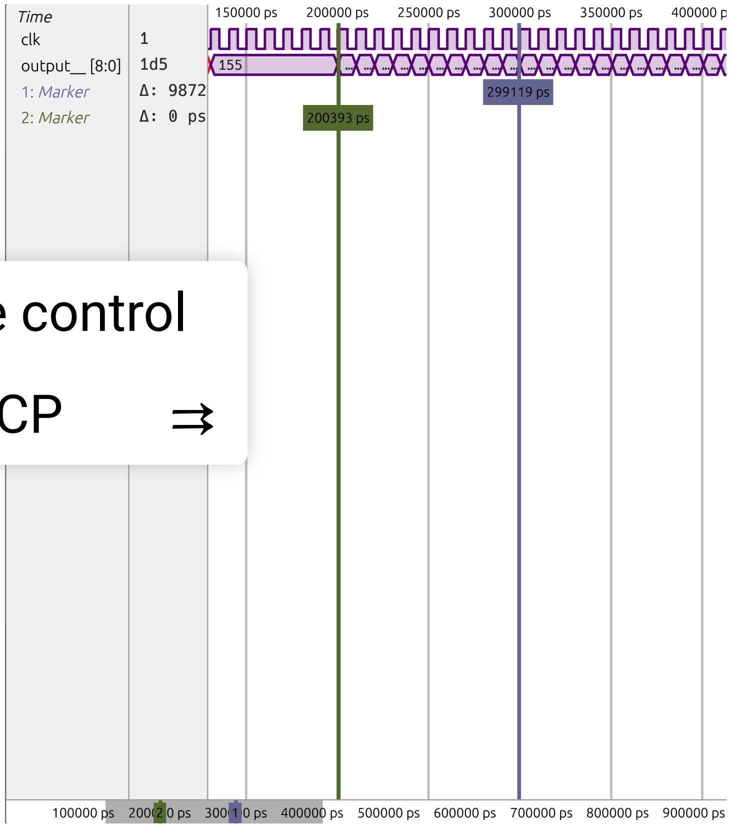
```
        set self.ready = uart_out.ready;
    }
}


entity udp_byte_burster(
    clk: clock,
    rst: bool,
    start: bool
) -> UdpStream {
    let ds = inst new_mut_wire();
    reg(clk) packets_left: uint<32> reset(rst: 0) = {
        if packets_left == 0 {
            if start {
                10
            } else {
                0
            }
        } else {
            if inst read_mut_wire(ds) {
                trunc(packets_left-1)
            } else {
                packets_left
            }
        }
    };

    let to_transmit = if packets_left != 0 {
        Some(
            UdpHeader$(
                source_port: None,
                destination_port: 1337,
                payload_length: 1460,
            ),
        )
    } else {
        None
    };

    // TODO: Just sending Some(0) constantly blocks the lower priority packets
    UdpStream(HeaderPayloadTxStream(Rv(&to_transmit, ds), Rv(&Some(0), port#1)))
}


```

```
NOR   hwtest/src/main.spade                                    1 sel   141:1
```





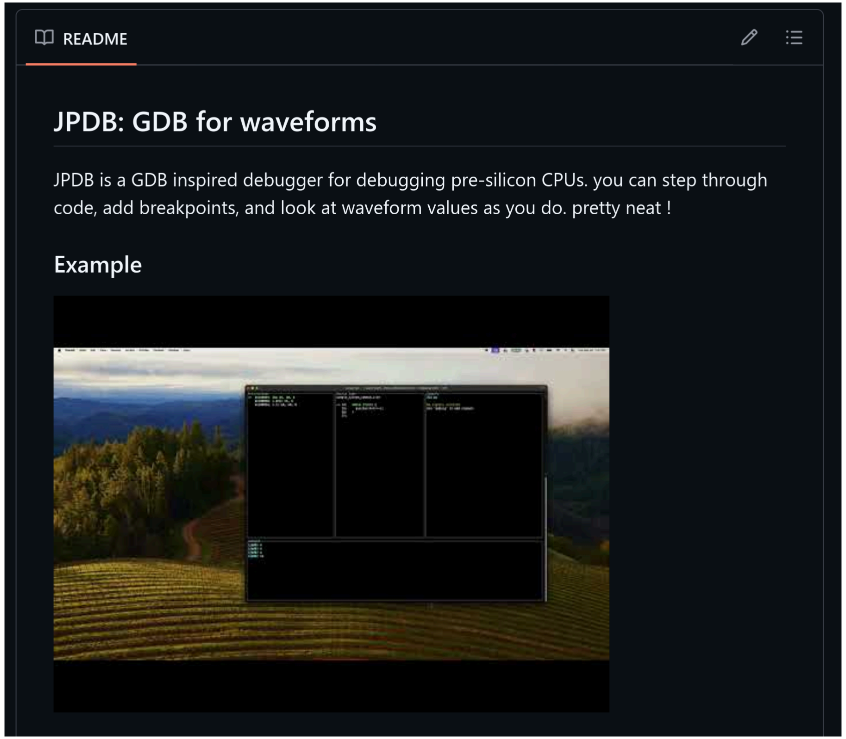
```
        set self.ready = uart_out.ready;
    }
}


entity udp_byte_burster(
    clk: clock,
    rst: bool,
    start: bool
) -> UdpStream {
    let ds = inst new_mut_wire();
    reg(clk) packets_left: uint<32> reset(rst: 0) = {
        if packets_left == 0 {
            if start {
                10
            } else {
                0
            }
        } else {
            if inst read_mut_wire(ds) {
                trunc(packets_left-1)
            } else {
                packets_left
            }
        }
    };

    let to_transmit = if packets_left != 0 {
        Some(
            UdpHeader$(
                source_port: None,
                destination_port: 1337,
                payload_length: 1460,
            ),
        )
    } else {
        None
    };

    // TODO: Just sending Some(0) constantly blocks the lower priority packets
    UdpStream(HeaderPayloadTxStream(Rv(&to_transmit, ds), Rv(&Some(0), port#1)))
}


 NOR   hwtest/src/main.spade                                    1 sel  141:1
```



Remote control

⇇ WCP ⇉

https://github.com/hudson-trading/slang-server



JPDB: GDB for waveforms

JPDB is a GDB inspired debugger for debugging pre-silicon CPUs. you can step through code, add breakpoints, and look at waveform values as you do. pretty neat !

Example

HDL Integration

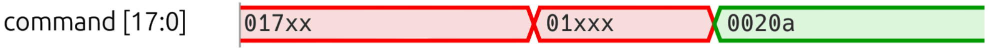
Spade

An HDL Inspired by modern software languages

HDL Integration

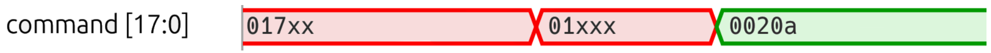
Spade

An HDL Inspired by modern software languages



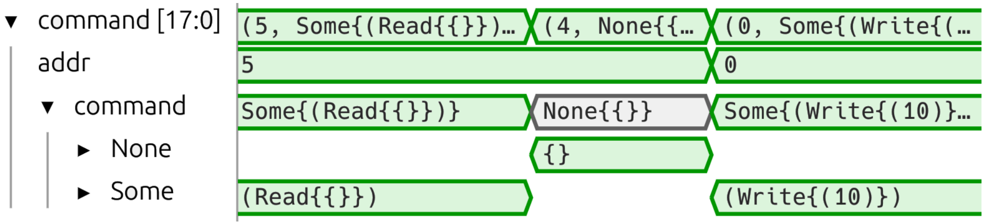
HDL Integration

```
struct Command {
  addr: uint<16>,
  inner: Option<InnerCommand>
}

enum InnerCommand {
  Read,
  Write{ data: T },
}
```



HDL Integration

```
struct Command {
  addr: uint<16>,
  inner: Option<InnerCommand>
}

enum InnerCommand {
  Read,
  Write{ data: T },
}
```



SILOGY

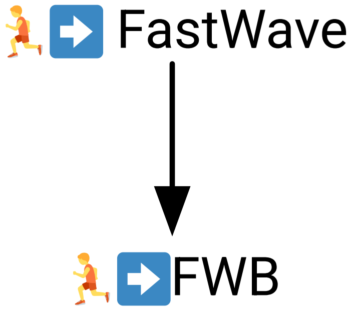
QS

LUBIS EDA

Customize

Collaborate

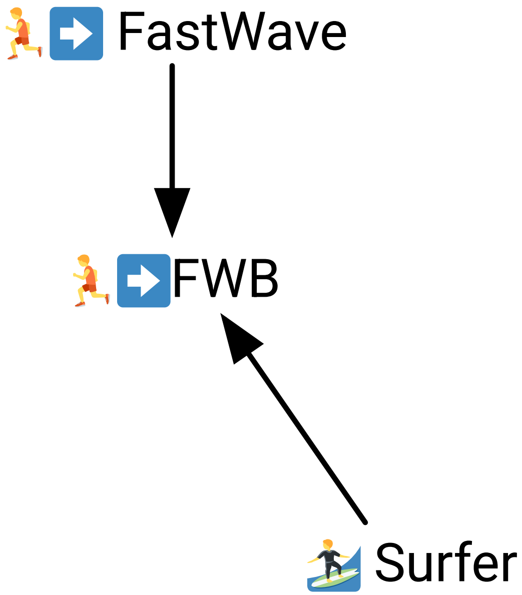
Waveform Backends



🏃➡️ FastWave

🏃➡️FWB

Waveform Backends

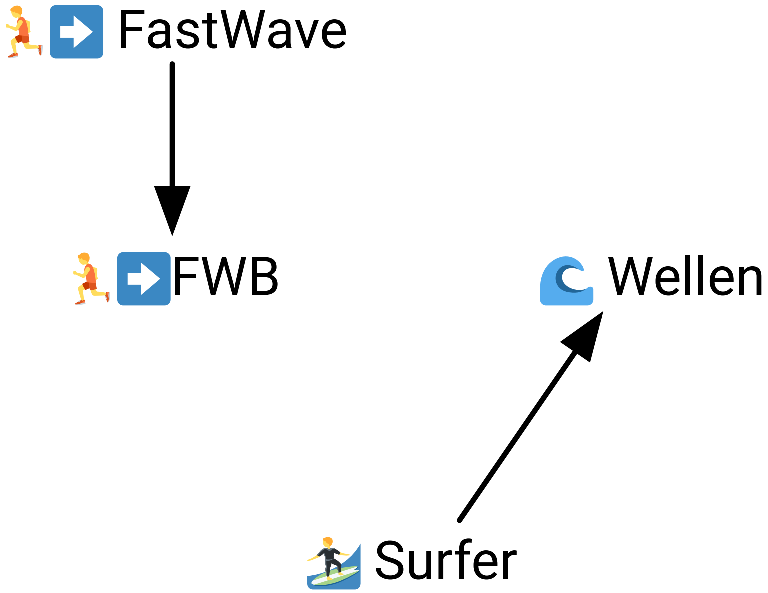


🏃➡️ FastWave

🏃➡️FWB

🏄 Surfer

Waveform Backends



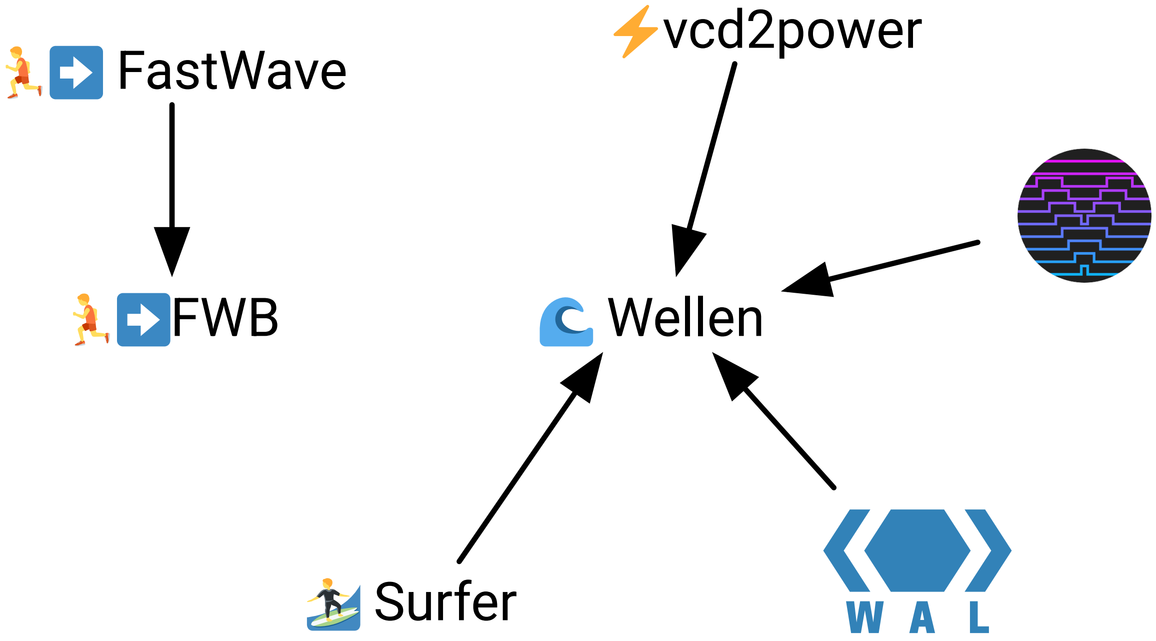
🏃➡️ FastWave

🏃➡️FWB

🌊 Wellen

🏄 Surfer

Waveform Backends



🏃➡️ FastWave

🏃➡️FWB

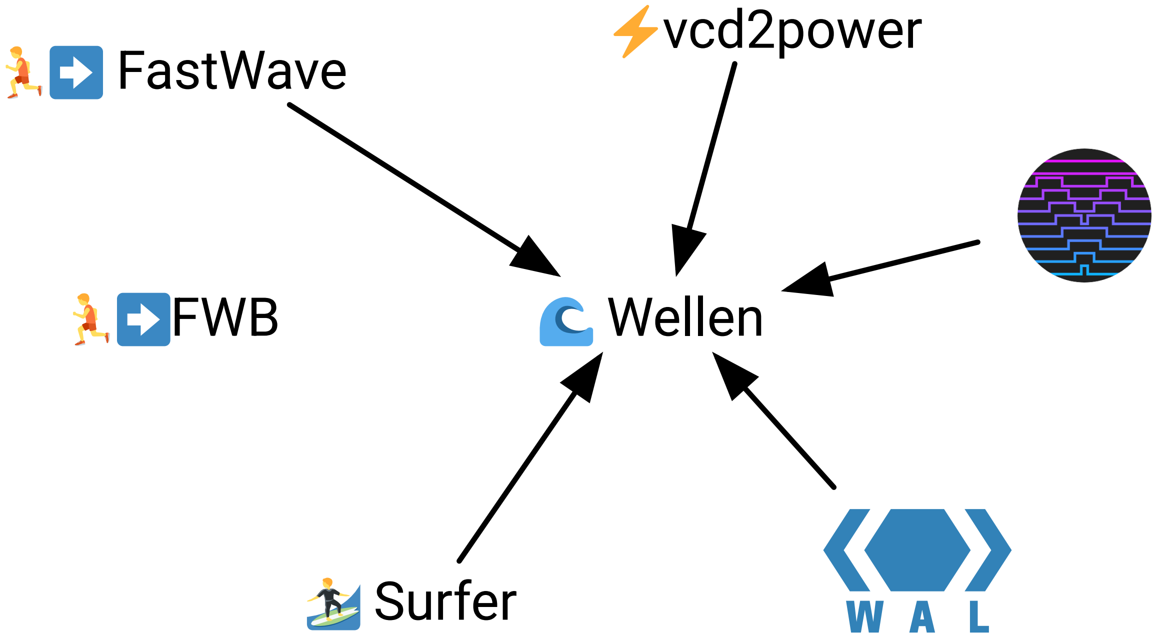
⚡vcd2power

🌊 Wellen

🏄 Surfer

WAL

Waveform Backends



FastWave

vcd2power

FWB

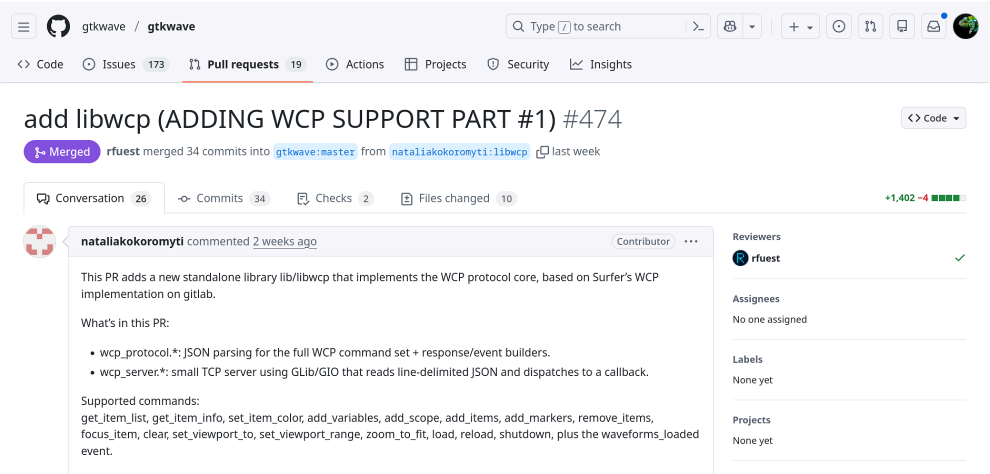
Wellen

Surfer

WAL

Waveform Control Protocol

Supported by all major waveform viewers



add libwcp (ADDING WCP SUPPORT PART #1) #474

Merged rfuest merged 34 commits into `gtkwave:master` from `nataliakokoromyti:libwcp` last week

nataliakokoromyti commented 2 weeks ago

This PR adds a new standalone library lib/libwcp that implements the WCP protocol core, based on Surfer's WCP implementation on gitlab.

What's in this PR:

- wcp_protocol.*: JSON parsing for the full WCP command set + response/event builders.
- wcp_server.*: small TCP server using GLib/GIO that reads line-delimited JSON and dispatches to a callback.

Supported commands:
get_item_list, get_item_info, set_item_color, add_variables, add_scope, add_items, add_markers, remove_items, focus_item, clear, set_viewport_to, set_viewport_range, zoom_to_fit, load, reload, shutdown, plus the waveforms_loaded event.

Other HDLs

Spade

Clash

Chisel

Re-use what you can, build what you need

Contact: frans.skarman@hm.edu, stefan.wallentowitz@hm.edu